Console Multiplexor

User Manual

Release 11

Avid Technology

Final Version

AUTOMATION SYSTEMS

Console Multiplexor
User's Manual
Release 11

BASYS Automation Systems, Inc.
5 Odell Plaza
Yonkers, NY 10701
U.S.A.
914-376-4800
Customer Support 1-800-869-7009
FAX 914-376-0865

BASYS Automation Systems, Ltd.
Langley Business Centre
11-49 Station Road
Langley
Berkshire
SL3 8YU
United Kingdom
44-(0)753-583333
After Hours 44-(0)272-270414
Telex 849778 BASYS G
FAX 44-(0)753-581559

BASYS Automation Systems, Inc.
7000 Peachtree Dunwoody Road
Atlanta, GA 30328
U.S.A.
404-396-6227
FAX 404-396-8752

© BASYS Automation Systems Inc. and Ltd. 1992

Final Version

BASYS News System is a trademark of BASYS Automation Systems, Inc. and Ltd.

All other product names referenced in this manual are trademarks of their respective owners.

While great care has been used in the production of this document, BASYS Automation Systems Inc. and Ltd. make no representation or warranties regarding the contents, except as set forth in the license agreement between BASYS Automation Systems Inc. and Ltd. and the licensee of this document. In no event will BASYS Automation Systems Inc. or Ltd. be liable for damages, including loss of profits, lost savings, or other incidental or consequential damages arising out of the use or misuse of this document, even if BASYS Automation Systems Inc. or Ltd. or an authorized representative of BASYS Automation Systems Inc. or Ltd. has been advised of the possibility of such damage.

BASYS Automation Systems Inc. or Ltd. reserves the right to revise this document or make changes to the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or change.

SOFTWARE LICENSE NOTICE

The software described in this document is under license from BASYS Automation Systems Inc. or Ltd. and may not be used or copied in any form whatsoever except in accordance with the terms of such license.

© BASYS Automation Systems Incorporated and Limited 1992
All rights reserved.
Reproduction of any portion of this document, in any medium, without the expressed written permission of BASYS Automation Systems Inc. or Ltd., is prohibited.

Final Version

Table of Contents

Preface: About This Manual

Reference v
Type of Computer v
DOS Essentials vi
Release 11 Changes vi
Screens vii
Keyboard Keys viii
Commands viii
Letters and Numbers x

Section One: Installation

Setting Up the Console Program

The Console 1-1
What You Need to Set Up the Console 1-2
Obtain a Backup Copy of the Console Program 1-3
Installing Console for the First Time 1-3
Upgrading Release 11 Console Onto a Hard Drive 1-4
Upgrading Console Onto a Single Floppy Drive 1-5
Upgrading Console Onto a Dual Floppy Drive 1-5
Finishing the Console Set Up 1-6
If You Have a Color Monitor 1-6
If You Don’t Have a Toshiba Keyboard 1-8
Starting the Console Program 1-9

Customizing the Dictionaries

Dictionaries 1-11
Videomap Dictionary 1-11
Keyboard Dictionary 1-11
Editing a Dictionary 1-11
Installing the Changes 1-12

Customizing the Keyboard

Hardware Scan Codes 1-15
VT200 Character Sequences 1-15
Assigning an ASCII Code to a Key 1-17
Guidelines for Editing the Keyboard Dictionary 1-18
CHART A 1-19
Keyboard Dictionary Descriptions 1-21
The Keyboard File 1-22

Final Version

Video Attributes

Video Attributes 1-25
Monochrome 1-26
Color 1-28

Character Mapping

IBM to MCS Character Mapping 1-31
MCS to IBM Character Mapping 1-33

Section Two: Usage

The Console

The Console Screen 2-1
The CMD Key 2-3
Selecting a Computer 2-3
Zoom 2-4
Which Computer to Select? 2-5
Select the Master Computer When 2-5
Select Any Computer When 2-5
Select All Computers When 2-5
Select a Specific Computer When 2-5
Console History 2-5
Viewing Recent History 2-6
Top 2-6
Bottom 2-6
Up 2-7
Down 2-7
List 2-7
Print 2-8
Sample Session 2-8
Reading Older History 2-9
Log Error 2-11
Changing Log Disks 2-12
Console Commands 2-12
Pausing the Screen Display 2-13
Function Keys 2-13
Assigning a Definition 2-14
Changing a Definition 2-14
If You Forget a Definition 2-15
Remote Console 2-15
Dialing In 2-15
Executing Commands Remotely 2-16
Logging Out 2-17
If You Can’t Get a Prompt on the Console 2-18
Exiting the Console 2-18
Starting the Console 2-19
Miscellaneous Notes 2-19

Final Version

Console Configuration File

Examining the Configuration File 2-21
Editing the Console Configuration File 2-22
Console Configuration Keywords 2-25
Computer 2-25
Hostess 2-25
Irq 2-25
Label 2-25
Log 2-26
Modem 2-27
Name 2-27
Password 2-27
Portaddress 2-27
Speed 2-27
Timeout 2-27

Console Superuser

Becoming a Console Superuser 2-29
Exiting Console Superuser 2-30
The Superuser Password 2-30

Section Three: Reference

Appendix A: Reference Charts

Appendix B: Computer Keyboards

Glossary

Index

×

Final Version

Final Version

Preface

The BASYS Console Multiplexor software (“console” for short) allows you to issue commands to aBASYS Automation Systems (B.A.S.) News System. The console software runs on an IBM-compatible personal computer (PC) that connects to your BASYS system so you can monitor and maintain your system. A modem line can be set up to allow a System Manager or B.A.S. representative to connect to and work on the console while he or she is away from the computer room.

If your News system is just being installed, a B.A.S. technician will set up the console program for you. If your system is already running, you will need to install any console upgrades yourself. The setup and configuration procedures for the console program are not difficult, but should be performed by a *System Manager*, who is responsible for setting up and maintaining the system.

This manual, written for System Managers, is divided into three, independent sections. Section 1 contains basic setup information. Section 2 contains information on using the console program. (If you’re interested in learning more about using the console program, you can go to Section 2 without reading Section 1.) Finally, Section 3 contains reference information, including charts, a glossary of terms and an index. The procedures you learn in this manual will prepare you to be able to perform the system maintenance steps explained in the News Operations Manual.

Reference

If you come across a word that you don’t understand, consult the glossary in Section 3. Terms that appear in the glossary are *italicized* when they are first mentioned. Use the index provided at the end of this manual for help in locating information on specific topics.

For your convenience, we have placed console reference information such as value tables and keyboard charts in the Reference section (Section 3).

Type of Computer

The console program can run on a PC that has a hard drive or one that just has floppy drives. Although there are instructions for installing or upgrading the console program for both types of computers, the rest of the manual assumes that the console program is being run off the A drive.

If the Console Program is on your Hard Drive
If you are running the console program off the hard drive, simply change the A drive to the C drive for all instructions.

Final Version

DOS Essentials

The installation and upgrade procedures discussed in this manual require a basic understanding of *DOS* (Disk Operating System) commands and structures. To get the maximum benefit from this manual, be sure you have a working knowledge of DOS before you go any further. Specifically, you should have an understanding of the following terms, commands, and concepts:

- Hierarchical File Structure
- Common DOS commands—**cd**, **checkdisk**, **copy**, **dir, type**, **xcopy**
- CONFIG.SYS
- **Edlin** (or another DOS file editor)

Release 11 Changes

Paragraphs that include information about features that are new or altered since Release 9 will have a vertical bar along the outside margin (such as the one alongside this paragraph). This is so you can identify new system changes as you read through the manual.

8-Bit Data Transmission

The console can now send and accept 8-bit data through a serial port. Because of this capability, the console's keyboard and video attributes can be customized. This capability also allows the console to send and receive international characters.

Although this represents a major change in the way the console program is set up, you will not notice extensive changes while running the program.

Additional Function Keys

Consoles that have enhanced keyboards have two additional function key definitions available (for a total of 14). The "+" and **ENTER** keys can now have assigned definitions. See "The Console" chapter for more information on assigning definitions to function keys.

Modem Security Enhancement

A "timeout" parameter can be added to the modem section of the configuration file to automatically log out a modem connection to the console after a specified period of inactivity. With this feature, the system can detect a modem that has been disconnected without a logout or a modem connection that has been inactive for a while. In either case, the system will log out the user and disconnect the modem, if necessary. See "Console Configuration File" chapter in Section 2 for more information.

Final Version

Log File Size Parameter

A “max size” parameter has been added to the log keyword in the configuration file. System Managers can use this parameter to specify the maximum size of disk log files. See the “Console Configuration File” chapter in Section 2 for more information.

Screens

This manual describes operations that you perform in DOS as well as operations you perform using the console program. To distinguish the DOS displays from the console displays, each is represented differently in this manual.

DOS Screens

The picture below shows how DOS commands will look on the console screen.

```
C:\>xcopy a:*.* c:\mux11 /s /e
Does MUX11  specify a file name
or directory name on the target
(F = file, D = directory)? d

Reading source file(s)...
```

Sample DOS Screen

The Console Screens

To distinguish the console screen displays from the rest of the manual, they are indented from the left margin and displayed in a different typeface, as shown below:

```
WBAS_A: su
password:            (enter password)
SU: so /dev/console
WBAS_A#
```

Console displays reproduced in this manual use bold italic to distinguish commands you enter from what the console displays on the screen. For example, in the case shown above, you would enter “su” at the “WBAS_A:” prompt.

The words to the side that are in parentheses (e.g., “enter password”) are instructions that tell you what to do when we cannot give you the exact entry to type at the console. (In the example above, we cannot tell you what to enter at the password prompt because that varies from system to system.) These words will not appear on your screen.

Note that when we show you how to edit a file using the DOS editor ***edlin***, the file will be shown in the same format as above, even though you will be in DOS. This will be the only time DOS screens will look like the console screens.

Final Version

Keyboard Keys

Often, you'll need to press a keyboard key to execute a command or to complete a step in a procedure. To indicate when the manual is referring to a keyboard key, the name of the key is displayed in upper-case, bold lettering. For example, **SHIFT** refers to the shift key and **ENTER** indicates the enter key.

Occasionally, you'll see a letter preceded by the "^" character (e.g., **^D**). The "^" indicates the **CTRL** key; when you see "^" in combination with another key, it means hold down the **CTRL** key while pressing the other key. For example, **^D** means type the **D** key while holding down the **CTRL** key.

Commands

In your work as System Manager, you'll use both DOS commands and console commands. The formats for each are described below. Note that *for the most part*, you'll be using DOS commands in the first section of the manual and console commands in the second section.

Executing a Command
To execute a DOS or console command, you must press **ENTER** after typing the command.

DOS Commands in Text

The names of commands you use in DOS are shown in lower-case, bold letters. For example, you might see the following: "use the **dir** command to see a listing of the directory's contents." The bold, lower-case letters indicate that "**dir**" is the name of a DOS command. We recommend keeping a DOS manual nearby for reference.

Console Commands in Text

Console commands, like DOS commands, are represented in this manual in bold, lower-case letters. For example, you might see a sentence like this:

> To restore the screen to its original state, use the **computer** command to select any computer (it doesn't matter which one you select).

Although both types of commands are represented similarly, you are not likely to confuse DOS commands and console commands since they are never used together.

DOS and Console Commands in Examples

When you are instructed how to type a command at the DOS prompt, the command and any arguments to the command are shown in lower-case, italicized lettering, as illustrated in the following example:

Final Version

DOS Commands

A:\DCT>*dir* **ENTER**

This is done to illustrate that a command does not need to be typed in upper-case. It also distinguishes the typed command from the prompts (e.g., **A:\DCT>**) and function keys (**CMD** and **ENTER**) which precede and follow the command.

When you are instructed to type a console command at the command prompt, the command and any arguments to the command are shown in lower-case, bold, italicized lettering, as illustrated in the following example:

Console Commands

`COMMAND` ***v console.cfg***

Note that, unlike DOS, the console software *is* case sensitive; therefore all the console commands must be typed in lower-case as they appear in the examples.

Be sure that your prompts look like the ones in our examples. (An exception is if the console program is on a hard drive. Then your DOS prompt will probably be something like "**C:\MUX11\DCT>**" instead of "**A:\DCT>**.")

General Formats

When introducing a command, we usually show the command's general format, including any *parameters* or *literals* that the command uses. A parameter is a value (such as, a file name) used with a command to modify what the command does. A literal is a word or a character that always has the same effect on the command. Parameters and literals may be required or optional depending on the command.

To distinguish parameters from literals and required parameters from optional ones, this manual uses the following convention: required literals are *not* bracketed, parameters are identified with angle brackets, and optional parameters and literals have straight brackets around them as shown below:

required literal
[optional literal]
<required parameter>
[<optional parameter>]

Note that parameters are *not* typed as shown. Since there is a choice of parameters, you would enter the parameter that you want to use for each specific example. Also note that this format applies to keywords in the configuration file. For example, when explaining the **log** keyword, the entry looks like the following:

log <drive:filename> [<computers>] [<max size>]

Final Version

The format description indicates that the line must begin with “log,” which is the required literal. In this example, there are no optional literals. Since the words “drive:filename” are in angle brackets, they are a required parameter, so a drive and filename must be entered in this position. The words “computers” and “max size” are in angle brackets within straight brackets, which indicates that they are optional parameters.

“Select a Computer”

Occasionally you’ll see phrases such as “select all computers” or “select the ‘A’ computer.” These phrases refer to the fact that your system uses more than one computer. Consequently, you will need to indicate which computer(s) you want a command to run on before executing it. You do this by *selecting* the computer(s) before running the command. Selecting a specific computer or all computers is covered in “Console” in Section 2.

The Console Prompt

The console displays a prompt for each computer based on the system ID and the computer’s name. Examples in this manual use a system that has “WBAS” (a fictional station) as its ID. Accordingly, the console prompts are derived from this ID. For example, the console prompt for the “A” computer will look like this:

```
WBAS_A:
```

More information about the kinds of prompts that will appear on the console screen can be found in “Console” in Section 2.

Letters and Numbers

A common mistake is to confuse the letters “l” and “o” with the numbers zero and one. Though you may be accustomed to typing the letter “o” on the typewriter to mean zero and the letter “l” to mean one, computers aren’t very imaginative and take everything you type literally. You must type the number “0” when you mean zero. Likewise, you must type the number “1” when you mean one.

In this manual, zeros appear as “0,” and the letter “o” looks like “o.” Unfortunately, the letter “l” looks very much like the number “1,” so wherever the potential for confusion exists, the text of the manual will make it clear whether it is referring to the letter or the number.

Spaces and Tab Spaces

When modifying site files, you’ll often need to enter one or more parameters on a line. Each parameter must be separated from the others by at least one space. You can create this space by pressing the **SPACE BAR** or by using the **TAB** key. We recommend that you choose one format or the other rather than switch between the two randomly.

Final Version

Final Version

Section 1
Installation

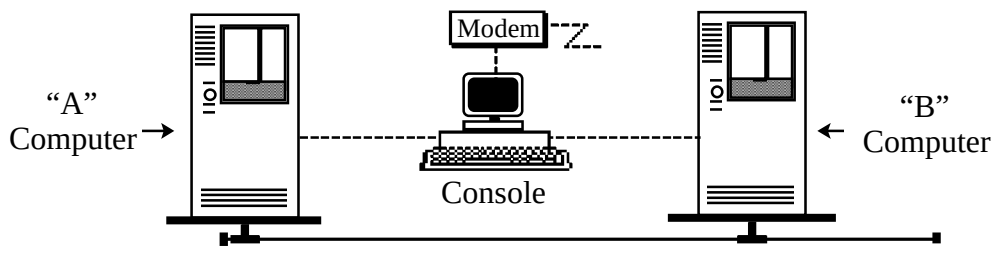
Final Version

Final Version

Setting Up The Console Program

Installing the Console Program

The Console Multiplexor program (“console”) is a DOS program that runs the *console* on your News System. The console is a personal computer (an IBM PC compatible) which connects directly to each computer in the system over a serial link and communicates with each computer simultaneously.



Console Connected to Host Computers

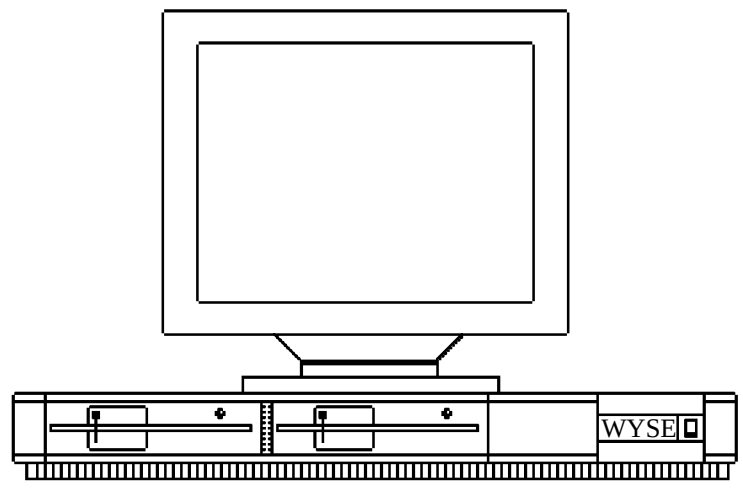
This chapter explains how to set up and start the console program. You will learn how to configure the program for your system, change the keyboard or video set-up (if necessary) and start the console program.

The Console

The console is one of the key components in your News system, and you’ll use it to do much of your System Manager work. System Managers use the console to enter commands, perform maintenance activities, and set system-wide options for the computer and for individual *users*.

The console software also provides for the use of a *remote console*—a personal computer or terminal that uses a modem to dial in to the console. The primary purpose of a remote console is to allow you or B.A.S. personnel to do routine maintenance and diagnostic work while away from the computer room. Access to the console through the modem is restricted with a password you assign. (Details on both the console and the remote console are available in “The Console” in Section 2.)

Final Version



Typical Console

What You Need to Set Up the Console

- IBM PC compatible computer with 8086, 8088, 80286, or 80386 CPU
- 640 Kb of memory
- A diskette drive that can read 3.5 inch 720Kb diskettes or 5.25 inch 360Kb diskettes (although a 1.2 megabyte diskette drive is highly recommended for the 5.25 inch format)
- 80-column video monitor, either monochrome or color
- MS-DOS version 3.3 or later
- CONFIG.SYS must specify the number of files to be nine or more

The shipped console diskette is set up to work best with the keyboard of a Toshiba 1000 or 1200 computer. However, if you have a different keyboard, we'll show you later how to change the keyboard style to more closely match the machine for which you are configuring the console.

To meet the configuration requirements involving the number of files, insert a DOS diskette into your A drive and turn the computer on. If you have a hard drive with DOS already on it, don't insert a diskette before you turn the computer on. When you get to the DOS prompt, use ***edlin*** (or some other DOS editor) to edit the CONFIG.SYS file as needed. In this case, you need to set the number of files to 9 or higher:

```
files = 9
```

If you make any changes to the system configuration, you'll have to reboot your computer to activate the new settings. If you want more detailed information on editing the CONFIG.SYS file, consult your DOS reference manual.

Once your console is set up properly and rebooted, you are ready to install the program.

Final Version

Obtain a Backup Copy of Console

Along with a working copy of the console program diskette, you should have at least one backup console diskette. This ensures that if something goes wrong with one copy of the console software, you have a functioning copy of the program to fall back on. Use the DOS **xcopy** command to make a backup of the program on a bootable diskette. Keep the backup copy in a safe, dry place.

Using the xcopy Command

Note that you must have DOS version 3.3 or greater to use the **xcopy** command. If you have an earlier version of DOS, you must first create a new directory and then copy the files into the directory. See your DOS manual for more information, if needed.

Your next step will be to install the console program.

If Your News System is Already Running

If your system has just been installed, the B.A.S. technician will probably set up the console program for you. However, it's a good idea for you to look over this section of the manual anyway, just to be familiar with the procedure.

If your system has been running with an older version of the console software, follow the upgrade instructions that begin on page 1-4.

Installing Console for the First Time

To begin the console software installation, place the console working diskette in drive A and type the following DOS command:

C:\>*xcopy a:*.* c:\mux11 /s /e* **ENTER**

You will get a message asking if "MUX11" specifies a file name or directory name. Type "d" for directory. This will create a MUX11 directory on the drive into which all the necessary files will be copied.

```
C:\>xcopy a:*.* c:\mux11 /s /e
Does MUX11  specify a file name
or directory name on the target
(F = file, D = directory)? d

Reading source file(s)...
```

Installing Console on the C Drive

The console software is now installed. You may need to modify the *CONSOLE.KEY* and *CONSOLE.CFG* files so that they are set up for your system's specific configuration. See

"Function Keys" and "Console Configuration File" in Section 2 for more information on what these files are and how to modify them.

Final Version

Now go to “Finishing The Console Set Up” (which starts on page 1-6) to complete the console software installation.

Upgrading Release 11 Console Onto a Hard Drive

When upgrading the console software on a hard drive, you will create a new directory to store the Release 11 files. You should choose a name for the directory that clearly identifies what files it contains. It is suggested that you use “MUX11” as the name of the new directory. Begin the installation by entering the command shown below.

C:\>*xcopy a:*.* c:\mux11 /s /e* **ENTER**

You will get a message asking if “MUX11” specifies a file name or directory name. Type “d” for directory. This will create a MUX11 directory on your hard disk into which all the necessary files will be copied.

Now you’ll probably want to copy your old CONSOLE.CFG and CONSOLE.KEY files to the new directory so that you won’t have to reenter any changes you had made to those files. (Note: in this example, the old console files are in the root directory.)

Before you copy the old files, change directories so that you are in the new directory.

C:\>*cd mux11* **ENTER**

Next, use the DOS **copy** command to copy the old CONSOLE.CFG and CONSOLE.KEY files to the new directory.

C:\MUX11\>*copy c:\console.cfg console.cfg* **ENTER**
C:\MUX11\>*copy c:\console.key console.key* **ENTER**

You will also need to change the AUTOEXEC.BAT file to make sure the new version of the console software is used when booting up the console or restarting the console program. Change the root directory’s AUTOEXEC.BAT file so that the “console” line reads “cd\mux11 console.”

Make Sure You Change the Root Directory File

It is important that you change the AUTOEXEC.BAT file in the root directory, and not the new file in the MUX11 directory. Unless you specify a special path in the CONFIG.SYS file, DOS will only look for the AUTOEXEC.BAT file in the root directory. Therefore, if you change the file in the MUX11 directory (without specifying the path in CONFIG.SYS), DOS will read the old AUTOEXEC.BAT file and start the older version of the console software.

You now have all the Release 11 console files on your hard drive. Follow the steps in “Finishing the Console Set Up” to complete the upgrade.

Final Version

Upgrading Console Onto a Single Floppy Drive

When you upgrade the console software, you'll probably want to keep your old CONSOLE.KEY and CONSOLE.CFG files so that you won't have to reenter any changes you had made to those files. To do that, you will need to copy the old files onto the new disk. To begin the upgrade, keep the old console software disk in the A drive and enter the following command:

A:\>*copy console.key b:\console.key* **ENTER**

You will get an "Insert diskette for drive B:" prompt. Change disks so that the Release 11 software disk is in the floppy drive, and then press any key to complete the copy. You will know that the file copy is finished when you see a "1 File(s) copied" message.

You will then be prompted to insert the diskette in drive A. Put the old console software disk in the drive.

Next, you will need to copy the CONSOLE.CFG file by entering the following command:

A:\>*copy console.cfg b:\console.cfg* **ENTER**

Once again, you will get a prompt telling you to change disks. Put the Release 11 software disk in the floppy drive and press any key to finish the copy. When you get the "1 File(s) copied" message, the file copy is complete. Even though the computer will prompt you to insert the old software disk in the drive, you don't have to change disks this time. Pressing any key will bring you to the A prompt.

You should now follow the steps in "Finishing the Console Set Up" to complete the upgrade.

Upgrading Console Onto a Dual Floppy Drive

When you upgrade the console software, you'll probably want to keep your old CONSOLE.KEY and CONSOLE.CFG files so that you won't have to reenter any changes you had made to those files. To do that, you will need to copy the old files onto the new disk. To begin the upgrade, keep the old console software disk in the A drive, put the Release 11 software disk in the B drive and enter the following command:

A:\>*copy console.key b:\console.key* **ENTER**

After you see the "1 File(s) Copied" message, enter the following command to copy the CONSOLE.CFG file:

A:\>*copy console.cfg b:\console.cfg* **ENTER**

When that file copy is complete, take the old disk out of the drive and put the new software disk in drive A.

Final Version

You should now follow the steps in “Finishing the Console Set Up” to complete the upgrade.

Finishing the Console Set Up

Now that the console program is set up on your console, you may need to customize the keyboard and/or screen set-up. If you have a monochrome monitor and a Toshiba 1000/1200 keyboard, continue with this section.

If you have a color monitor or a keyboard other than the Toshiba 1000/1200, skip the next step and move on to “If You Have a Color Monitor” or “If You Don’t Have a Toshiba Keyboard.”

Once you have all the Release 11 files on the A or C drive, you’ll need to run a program called ***makemux***. This program installs the contents of all the dictionary files into the console program. You’ll first have to change directories so that you are in the DCT *sub-directory* when you run the **makemux** program. To run **makemux**, you must include a space, two periods, another space, a minus sign and the letter “i” after the command. The command should look like the following:

A:\DCT>*makemux .. -i* **ENTER**

The computer will translate all the dictionaries while it sets up the program. Your console is now ready to send commands to your computers and display messages.

You can skip ahead to the section called “Starting the Console Program” at the end of this chapter.

If You Have a Color Monitor

The console software comes configured to work on a monochrome monitor. If you have a color monitor, you can specify that you want the console to use color. If you try to run the console software on a color monitor without the color dictionary installed, you may find the text hard to read. The color dictionary file is intended to provide a more legible display on a color monitor.

Use the DOS **dir** command as shown to get a listing of the disk’s contents. This should look something like the example below.

Final Version

```
A:\>dir

 Volume in drive A is MUX 11
 Directory of  A:\

CONSOLE   EXE        47155     1-01-80   6:23p
CONSOLE   CFG         1073    11-25-91   4:26p
CONSOLE   KEY          116    11-25-91   4:26p
AUTOEXEC  BAT           89    11-25-91   4:26p
LOG       TXT          155    12-04-91   3:52p
DCT             <DIR>        12-04-91   3:52p
EDLIN     COM        7183     8-14-84   8:00a
          7 File (s)      194560   bytes free

A:\>_
```

Root Directory Listing

Next, go into the DCT sub-directory and use the **dir** command again to get a listing of DCT's contents. It should be similar to the listing below.

```
A:\>cd dct
A:\DCT>dir
 Volume in drive A is MUX 11
 Directory of A:\DCT

.                <DIR>        12-04-91   3:54p
..               <DIR>        12-04-91   3:54p
MAKEMUX   EXE        28171 11-25-91   4:10p
KEYBOARD  DCT         7645 11-25-91   4:26p
VIDEOMAP  DCT         4691 11-25-91   4:26p
DEFAULT   KBD         7516 11-25-91   4:26p
IBM       KBD         6513 11-25-91   4:26p
IBMX      KBD         6579 11-25-91   4:26p
TOSHIBA   KBD         7657 11-25-91   4:27p
ZENITH    KBD         7311 11-25-91   4:27p
MONO      VID         4688 11-25-91   4:27p
COLOR     VID         5092 11-25-91   4:27p
         12 File (s)      194560 bytes free

A:\DCT>_
```

DCT Directory Listing

Notice the files MONO.VID and COLOR.VID (at the bottom of the directory listing). These are the two dictionaries that are provided with the console program to define your video display. Since MONO.VID (monochrome) is the current assignment, you'll need to copy COLOR.VID to the video dictionary file that the console software uses as a reference. This dictionary file, located near the top of the directory listing, is called VIDEOMAP.DCT. To assign color, type the following command:

A:\DCT>*copy color.vid videomap.dct* **ENTER**

Final Version

Once this is done, you'll need to run a program called ***makemux***. This program installs the new videomap descriptions into the console software. When running the program, you

must include a space, two periods, another space, a minus sign, and the letter "i" after the command, as follows:

A:\DCT>*makemux .. -i* **ENTER**

You will see the computer translate the videomap and keyboard dictionaries. Because **makemux** automatically reads both dictionaries, you can change both the videomap dictionary and the keyboard dictionary (as described next) and then run the **makemux** program only once. For more information on the difference between MONO.VID and COLOR.VID, see the "Video Attributes" chapter later in this section.

You're done. You can now run the console program on a color monitor.

If You Don't Have a Toshiba Keyboard

The console software uses the Toshiba 1000/1200 keyboard as its default. If you have another type of keyboard, you may be able to customize the console software to work with your particular keyboard. This is possible if you are using an IBM standard keyboard, an IBM extended keyboard, or a Zenith SUPERSPORT™ laptop.

Just as you changed the videomap dictionary to address a color monitor, you can change the keyboard dictionary to address a different keyboard. For example, let's say you have an IBM extended keyboard. Notice the files KEYBOARD.DCT and IBMX.KBD in the DCT sub-directory. You will copy the IBMX.KBD descriptions into the keyboard dictionary:

A:\DCT>*copy ibmx.kbd keyboard.dct* **ENTER**

When this is complete, you'll need to run **makemux**. This will install the new keyboard descriptions into the console software. Don't forget to type a space, two periods, another space, a minus sign, and the letter "i" after the command.

A:\DCT>*makemux .. -i* **ENTER**

You're done. You can now run the console software with an IBM extended keyboard.

If you have one of the following types of keyboards, you can copy the corresponding alternate dictionary to the keyboard dictionary in the way just illustrated.

Keyboard Style	**Alternate Dictionary**
Toshiba T1000 or T1200	toshiba.kbd
IBM PC XT Compatible	ibm.kbd
IBM PC AT Compatible	ibmx.kbd
Zenith SUPERSPORT	zenith.kbd

Final Version

If you want to customize one of these keyboards to better suit your needs, you will find the keyboard drawings in Appendix B helpful after reading “Customizing the Keyboard” later in this section. If you are using a keyboard for which we do not provide an alternate keyboard file, refer to the chapter called “Customizing the Keyboard.”

Starting the Console Program

After the dictionaries are set up, you can start the console program. To do that, make sure you’re in the MUX11 directory if the console software is on a hard drive or in the root directory if you’re using console from a floppy drive. Type “console” at the DOS prompt as shown below and press **RETURN**.

A:\>*console* **ENTER**

After a few seconds, you’ll see the console screen, partitioned into separate regions for each computer in the system. See Section 2 for instructions on how to use the console program.

If You Get International Characters

Occasionally you may see some international characters mixed in with a login prompt. Simply enter whatever you would normally type and then press **RETURN**. After you press **RETURN**, the characters will all be correct.

There is nothing wrong with your system. This happens because UNIX is sending international characters. Therefore, that is what shows up on your screen. This may only occur in a few circumstances, including occasionally when you restart the console program.

×

Final Version

Final Version

Customizing the Dictionaries

There are two DOS files that contain *translations* for international characters, descriptions for video attributes and descriptions for individual keys. These files, called *dictionaries*, are VIDEOMAP.DCT and KEYBOARD.DCT. The console software has two of these dictionaries, which describe international characters and individual keys. For example, the key descriptions are in a dictionary called “KEYBOARD.DCT.”

Dictionaries give you the freedom to customize the console’s keyboard. For example, you can add international characters to your keyboard by editing their descriptions in the videomap dictionary and then installing the modified dictionary into the console program.

This chapter will show what each of the console’s dictionaries describes and how to modify these dictionaries and install your changes into the console program.

Dictionaries

The console program uses two separate dictionaries. The following table lists their names and the files where they are stored. The contents of the two dictionaries are listed below and then discussed in greater detail in subsequent chapters.

Dictionary Name	Filename
Video Attributes	VIDEOMAP.DCT
Key Assignments	KEYBOARD.DCT

Videomap Dictionary

This dictionary, stored in VIDEOMAP.DCT, contains the translations for international characters such as £ and å, as well as descriptions for video attributes. The chapter called “Video Attributes” contains a more detailed description of this dictionary and provides instructions for making changes to it.

Keyboard Dictionary

This dictionary, stored in KEYBOARD.DCT, contains the descriptions for individual keys. The chapter called “Customizing the Keyboard,” which follows this chapter, contains detailed descriptions of this dictionary and provides instructions for making changes to it.

Final Version

Editing a Dictionary

To change a key description, you must edit the appropriate dictionary. For example, if you want to change the description of data key 14 of the IBM extended keyboard from

delete (“<del>”) to backspace (“<bs>”), you must first open IBMX.KBD for editing as shown below.

A:\DCT>*edlin ibmx.kbd* **ENTER**

After opening this file, move to the line containing the description for key 14. Next, use the **RIGHT ARROW** key to move the cursor to the “d” in the first “del” description (which is in the third column). Type “bs> ” (don’t forget the space after the right angle bracket). Then press the **RIGHT ARROW** key until you get to the “d” in the second “del” description (in the fourth column).

Continue changing “del” to “bs” until all four “delete” descriptions have been changed to “backspace.” To keep <nul> as part of the key description, press the **RIGHT ARROW** key until you reach the end of the line. (Note that you can also press the **F3** key after changing the last “del” description and **edlin** will automatically finish the line for you.) After you’ve made your changes and pressed **ENTER**, type “e” to save the changes and exit from **edlin**.

```
* 20
     20:*data  14   <del>      <del>      <del>      <del>      <nul>
     20:*data  14   <bs>       <bs>       <bs>       <bs>       <nul>
* e
```

Changing a Key Description From “del” to “bs”

Before you can use the new key description, you need to copy the IBMX.KBD file into KEYBOARD.DCT.

A:\DCT>*copy ibmx.kbd keyboard.dct* **ENTER**

Next, you’ll need to install the changes into the console program as described below.

Installing the Changes

When the console software was first installed, the **makemux** program was executed to read and install the dictionaries. After modifying a dictionary, you must again run **makemux** to have it read the modified dictionaries and install your changes into the console program.

From within the DCT sub-directory, issue the following command, being sure to include two periods, a space and a “-i” before pressing **ENTER**:

A:\DCT>*makemux .. -i* **ENTER**

Final Version

If you do not enter the command correctly, a usage message will be displayed to show the proper syntax for invoking the **makemux** program with an explanation of each option.

```
A:\DCT>makemux
MAKEMUX directory [-iskvD]
        'i'  install the descriptions (otherwise, just test)
        's'  use the standard descriptions
        'k'  show keyboard descriptions
        'v'  show videomap descriptions
        'D'  is diagnostic (echo descriptions)

the directory must contain "console.exe"

MAKEMUX  # (@)  11.09.10

A:\DCT>_
```

List of Makemux Options

As shown above, there are five options that can be entered with the **makemux** command. The only time dictionary changes will be installed, however, is when the “-i” option is entered. Following is a brief description of the other options.

The “-s” option is used to reinstall the standard descriptions, which are the descriptions the console software had in memory when you got the disk. Entering “-s” alone will not do anything. You need to enter “-si” to install the standard descriptions.

The other options are used for diagnostic purposes only. When you enter the “k” or “v” options, **makemux** will print out what it thinks that dictionary’s descriptions are. For example, when you enter the “makemux .. -k” command, **makemux** will print each key description as it is processed. (The printed dictionary will scroll off the console screen quickly, so you will probably want to write it to a DOS file and review it after the command has finished executing.) If you had previously made any modifications to the keyboard dictionary, you would see the changes; however, the modifications will not be installed until you enter the “makemux .. -i” command. The “v” option will similarly print the videomap dictionary.

Note that you can combine the options. For example, you can enter “makemux .. -kv” to have both the keyboard and videomap dictionaries printed. You can also have both dictionaries printed by entering the **makemux** command with the “-d” option.

The two periods used in the command tell DOS where to find console’s executable file, CONSOLE.EXE. In this case, the command tells DOS that the CONSOLE.EXE file is in the parent directory of the current directory (DCT). That is, CONSOLE.EXE is in the root directory.

While **makemux** is running, it will display messages about which dictionary it is currently translating and how much space is still available for the descriptions, if any.

Final Version

Once **makemux** is finished, you can restart the console. After restarting the console, try using the **BACKSPACE** key. It now will delete the character to the left of the cursor instead of the character under the cursor.

“Table Space Exceeded”

If a set of dictionaries exceeds the amount of space allotted to it, **makemux** will display a message indicating by how many characters the descriptions have exceeded the available space. If this happens, you’ll need to remove the extra characters.

For more information on the keyboard dictionary and how to customize it, see the next chapter, “Customizing the Keyboard.”

×

Final Version

Customizing the Keyboard

Editing the Keyboard Dictionary

To customize your keyboard, you'll have to edit the codes and strings in the keyboard dictionary that correspond to the key you are customizing and then run **makemux** to incorporate the changes. In this chapter, we'll discuss the different options you have for defining keys in the keyboard dictionary.

Hardware Scan Codes

Each key on an IBM compatible computer is identified by the internal microprocessor with a *hardware scan code*. The key labeled **F1**, for instance, generates the scan code 59, regardless of what kind of PC-compatible machine you are using. At the end of this chapter you'll find Chart A, which shows you the scan codes sent by each of the keys commonly found on IBM-type keyboards.

Each key on the keyboard generates a unique scan code. Even if the same label appears on two keys (such as on the number labels that appear in the alphanumeric section of a typical keyboard as well as in the numeric keypad) the individual keys always generate their own codes. The scan code for a given key is the same whether **CAPS LOCK** or some other modifier key (such as **SHIFT**, **CTRL**, or **ALT**) is in effect or not.

The keyboard dictionary uses the hardware scan code to identify individual keys. The codes and sequences listed beside the scan code determine how each key will function. Next, we'll discuss the codes and sequences that can be assigned to each key.

VT200 Character Sequences

One option for what can be assigned to a hardware scan code in the keyboard dictionary is a *VT200 character sequence*. A VT200 character sequence is a code used to correspond to each function key on a VT200-style terminal keyboard, just as scan codes correspond to each key on a PC keyboard. The console, which is a PC and not a VT200 terminal, needs a facility to assign VT200 character sequences to individual hardware scan codes. The keyboard dictionary effectively converts a PC keyboard into a VT200-style keyboard.

VT200 character sequences are assigned to hardware scan codes in the console *keyboard dictionary*, which can be found in DOS in the DCT sub-directory. Go to the DCT sub-directory now and use the DOS **type** command to examine the file called KEYBOARD.DCT.

Following is an excerpt of the installation keyboard dictionary that comes with console:

Final Version

```
data      50    m             M             M             <CR>          <nul>
data      51    ,             ,             <<            <nul>         <nul>
data      52    .             .             >             <nul>         <nul>
data      53    /             /             ?             <nul>         <nul>
shift     54
data      55    <ESC>OD       <ESC>OD       <ESC>OD       <ESC>OD       <nul>
data      56    <ESC>OM       <ESC>OM       <ESC>OM       <255>         <nul>
data      57    <SP>          <SP>          <SP>          <nul>         <nul>
capslock  58
data      59    <ESC>OQ       <ESC>OQ       <ESC>OQ       <ESC>OQ       <nul>
data      60    <ESC>OR       <ESC>OR       <ESC>OR       <ESC>OR       <nul>
data      61    <ESC>OS       <ESC>OS       <ESC>OS       <ESC>OS       <nul>
data      62    <ESC>Ov       <ESC>Ov       <ESC>Ov       <ESC>Ov       <nul>
data      63    <ESC>Om       <ESC>Om       <ESC>Om       <ESC>Om       <nul>
data      64    <ESC>Oq       <ESC>Oq       <ESC>Oq       <ESC>Oq       <nul>
data      65    <ESC>Ot       <ESC>Ot       <ESC>Ot       <ESC>Ot       <nul>
data      66    <ESC>[3~      <ESC>[3~      <ESC>[3~      <ESC>[3~      <nul>
data      67    <ESC>OP       <ESC>OP       <ESC>OP       <ESC>OP       <nul>
data      68    <ESC>Ow       <ESC>Ow       <ESC>Ow       <ESC>Ow       <nul>
data      69    <ESC>OA       <ESC>OA       <ESC>OA       <ESC>OA       <nul>
```

KEYBOARD.DCT File Excerpt

Notice that the keyboard dictionary file is laid out as a chart with rows and columns. The first column describes the type of key being programmed; that is, whether the key is associated with variable information (“data”), or some permanent key function (i.e., “capslock,” “shift,” “control,” or “alternate”). In the second column from the left are the hardware scan codes for each key. To the right of each key’s scan code are the VT200 character sequences or other codes that are assigned to each *shift state* of that key. By s we mean the way that each key is pressed—either alone or in combination with another key that may alter its function.

Each key has five possible shift states:

3rd Column	*Unshifted or Normal*	when the key is pressed by itself with no other key pressed in conjunction with it.
4th Column	*Caps Lock*	when the key is pressed while the **CAPS LOCK** key is engaged.
5th Column	*Shifted*	when the key is pressed while the **SHIFT** key is engaged.
6th Column	*Control*	when the key is pressed while the **CTRL** key is engaged.
7th Column	*Alternate*	when the key is pressed while the **ALT** key is engaged.

Shift states cannot be added together or combined. For instance, there can be no **CTRL-CAPS LOCK** state of a key. If you press more than one shift key at a time, the order of precedence is: **ALT** (highest), **CTRL**, **SHIFT**, **CAPS LOCK** (lowest). That is, **ALT**

Final Version

overrides any other shift state, **CTRL** overrides every other shift state except **ALT**, and so on.

You can assign different VT200 character sequences, displayable characters, or *ASCII codes* to each of these shift states, or you can specify that the same sequence, character, or code is invoked no matter which shift state of the key is used.

For example, notice that the key with the scan code 50 has been programmed to generate a lowercase “m” when unshifted, and an uppercase “M” when shifted or when **CAPS LOCK** is engaged. However, when pressed in conjunction with the **CTRL** key (6th column), that key will generate a carriage return (“<CR>”), and will do absolutely nothing (“<nul>”) when pressed in conjunction with the **ALT** key (7th column). The key with the scan code 59, on the other hand, will generate the same thing (“<ESC>OQ”) no matter which shift state is applied to the key, except for the alternate shift state, in which case nothing will happen—(“<nul>”).

Note that the VT200 character sequences are the items that usually begin with “<ESC>”, but “<del>”, “<lf>”, and “\” are also VT200 character sequences.

The Translation Process

When you press a key, a unique hardware scan code is generated. Each scan code has a VT200 character sequence, displayable character or ASCII code assigned to it in the KEYBOARD.DCT file. That sequence or code is then translated by the B.A.S. News software which tells console what to do. The console then performs the action.

For example, with the IBM extended keyboard, pressing the **ENTER** key at the bottom right of the keyboard generates scan code 101. The console program then looks at the KEYBOARD.DCT file and checks to see what VT200 character sequence is associated with scan code 101.

```
data    100    <ESC>OQ      <ESC>OQ      <ESC>OQ      <ESC>OQ    <nul>
data    101    <ESC>OM      <ESC>OM      <ESC>OM      <ESC>OM    <nul>
data    102    <ESC>Ot      <ESC>Ot      <ESC>Ot      <ESC>Ot    <nul>
```

KEYBOARD.DCT for an IBM Extended Keyboard

As you can see in the picture above, scan code 101 generates the sequence “<ESC>OM” which is defined by B.A.S. News software as the sequence for **COMMAND**. Therefore, when you press the **ENTER** key on an IBM extended keyboard, a “COMMAND” prompt will be generated at the console.

Assigning an ASCII Code to a Key

Hardware scan codes in the keyboard dictionary may have ASCII codes or displayable characters (such as letters) assigned to them rather than VT200 character sequences. When a key’s scan code is assigned an ASCII code or displayable character, the key customization is complete; that key will perform the function that is assigned to it in the

Final Version

dictionary and the key assignment will be sent directly to the remote console during a connection without any further processing by console.

For example, when a user presses the **SPACE BAR**, scan code 57 is generated. Console then looks up the description for scan code 57 in the KEYBOARD.DCT file. Scan code 57 is assigned "<SP>", which is the ASCII code for "space." When the console program sees "<SP>" it will automatically "print" a space. The translation is done; the code doesn't have to be processed by the B.A.S. News software.

For a guide to some ASCII codes that are available, check the partial ASCII chart in Appendix A.

Other Options and Guidelines for the Keyboard Dictionary

Besides VT200 character sequences and ASCII codes, there are other codes that can be assigned to the PC's keys—*compose sequences*, for instance. For a discussion of some of these options, as well as a few other guidelines on how to assign descriptions to keys, check the discussion of other keyboard descriptions at the end of this chapter.

Guidelines for Editing the Keyboard Dictionary

You can use a DOS editor such as **edlin** to make changes to the keyboard dictionary. Follow the general procedure discussed in the previous chapter under "Editing a Dictionary," but keep the following considerations in mind.

Edit the Dictionary According to Scan Code, NOT Line Number

Note that when you edit the dictionary, the line numbers that appear on the far left of the screen may closely correspond with the hardware scan code number on that line, but they *aren't* the same. Be careful to edit the line according to the proper scan code and not the line number.

Once you edit the keyboard dictionary, you have to run the **makemux** program in DOS, which will install the changes you've made to the dictionary. From within the DCT sub-directory, you can issue the following command:

A:\DCT>*makemux .. -i* **ENTER**

You can now run the console program with the changes you made.

Final Version

CHART A

Correspondence Between Key Labels and Hardware Scan Codes on IBM Keyboards

Alphanumeric Key Area

Key Label	Hardware Scan Code	Key Label	Hardware Scan Code
A	30	3 or #	04
B	48	4 or $	05
C	46	5 or %	06
D	32	6 or ^	07
E	18	7 or &	08
F	33	8 or *	09
G	34	9 or (	10
H	35	` or ~	41
I	23	- or _	12
J	36	= or +	13
K	37	[or {	26
L	38	] or }	27
M	50	\ or |	43
N	49	; or :	39
O	24	' or "	40
P	25	, or <	51
Q	16	. or >	52
R	19	/ or ?	53
S	31	Shift (left)	42
T	20	Ctrl (left)	29
U	22	Alt (left)	56
V	47	Shift (right)	54
W	17	Ctrl (right)	103
X	45	Alt (right)	102
Y	21	Caps Lock	58
Z	44	Backspace	14
0 or)	11	Tab	15
1 or !	02	Spacebar	57
2 or @	03	Enter	28

Final Version

Numeric/Cursor Keypad

Key Label	Hardware Scan Code
Num Lock	69
/	100
*	55
-	74
+	78
Enter	101
1 or End	79
2 or ↓	80
3 or Pg Dn	81
4 or ←	75
5	76
6 or →	77
7 or Home	71
8 or ↑	72
9 or Pg Up	73
0 or Ins	82
. or Del	83

Function Keys

Key Label	Hardware Scan Code
Esc	1
F1	59
F2	60
F3	61
F4	62
F5	63
F6	64
F7	65
F8	66
F9	67
F10	68
F11	87
F12	88
Print Screen	84
Scroll Lock	70
Pause	89

Final Version

Dedicated Cursor Area and Related Keys

Key Label	Hardware Scan Code
Insert	90
Home	91
Page Up	92
Delete	93
End	94
Page Down	95
Up arrow	96
Left arrow	97
Down arrow	98
Right arrow	99

Keyboard Dictionary Descriptions

In addition to VT200 character sequences and ASCII codes, there are other special notations that can be used in key descriptions. Following are a few considerations that you should be aware of when configuring the console keyboard dictionary.

When you want to assign a displayable character (such as a letter) to a key, you can simply type that character into the dictionary on the line that corresponds to the desired key's scan code. There are a couple of exceptions, however. A semicolon, although a displayable character, is used in the dictionary to signal that what follows is a comment that is to be ignored by the computer. If you want to assign a semicolon to a key on your PC keyboard, you'll have to type "<sc>" instead of a ";" character.

Also, to generate a "<" character, you must enter "<<" in the dictionary, because a single "<" serves as an introduction to a translation sequence. A similar condition holds true for the underline character, ("_"). You'll need to type the character twice in the dictionary to generate the character from a key since a single "_" serves as an introduction to a DEC compose sequence (see below).

Non-displayable, non-printable can also be assigned in the dictionary. These codes correspond to specific functions like carriage ("<cr>") and line ("<lf>") which aren't displayed, but which perform an action instead. Check the partial ASCII chart in Appendix A at the end of this manual for other non-printable ASCII codes that can be used as key descriptions.

It is also possible to enter compose sequences into the keyboard dictionary. Compose sequences are key combinations that generate characters that do not normally appear on a keyboard, but which can still be displayed on the screen. Examples include à, Ÿ, £, and ¥. These are known as *extended* or *compose characters*, and they are discussed in more detail in the "Character Mapping" chapter. The characters that are available for assignment in the keyboard dictionary come from what is known as the *DEC Multinational Character Set*. Their corresponding character sequences can be found in the chart in Appendix A.

Final Version

Briefly, by assigning a compose sequence to a key in the keyboard dictionary, you are getting that key to generate a character that normally doesn't appear on a keyboard. If, for example, you wanted to display the yen symbol at the touch of a key, you could assign the character sequence that generates "¥" to the scan code of the key you desire. The proper sequence can be determined by checking the international character description chart in Appendix A. Compose sequences begin with an underscore character ("_") which is followed by two more characters. For example, you would enter the sequence "_y=" as the key description that would generate the "¥" character.

As an option, instead of entering the character , you could use the actual character as the key description *if* you know how to generate that character on your console's screen. Information on generating particular characters may be found in your PC's manual.

If you want to be able to generate compose characters on your own, you can define a key as the *compose key* by using the description "<255>." The compose will allow you to enter the compose sequence directly from the keyboard.

To compose an extended character, you need only press the key defined as "<255>" and then the two final characters of the desired character sequence. The proper sequence can be determined by checking the chart called "International Character Descriptions" in Appendix A. Remember that the compose key replaces the "_" as the first character in each sequence when the user composes the character.

To generate the "¥" character, for instance, you would press the key defined as the compose key, then the "y" key, then the "=" key. When this sequence is complete, the "¥" character will appear on the screen. If you press an invalid sequence, the computer will beep and you must begin the sequence again, pressing the compose key first.

The Keyboard File

You may have noticed that the beginning of the installation keyboard file looks something like that shown below:

```
enhanced        1
devatt          <nul>
answer          vt100<cr> <lf>

;TYPE    SCAN    NORMAL      CAPSLOCK       SHIFT       CONTROL              ALT
data      1      <ESC>       <ESC>          <ESC>       <ESC>                <nul>
data      2      1           1              !           <255> <254> <1>      <nul>
```

KEYBOARD.DCT File Excerpt

The three lines at the top ("enhanced," "devatt" and "answer") have to do with what type of computer set-up you have. "Enhanced" has to do with your keyboard, while "devatt" and "answer" are strings that the console may send to a remote system if someone has established a remote login.

Final Version

A “normal” keyboard has 83 scan codes. Some PCs, however, have extended keyboards that can generate 103 scan codes. If the dictionary has “enhanced 0” at the top, that means the keyboard is not enhanced. “Enhanced 1,” on the other hand, means that the keyboard is extended and so the dictionary includes descriptions for scan codes 84-103. Note that a keyboard dictionary may include 103 scan code descriptions even if it is not enhanced; the extra descriptions simply will not be used.

Sometimes when someone is connected to the console via a modem, the remote computer will send an enquiry to the console. The two strings listed with “devatt” and “answer” are the console’s responses to the remote’s questions. For example, the remote computer may ask the console if it has any special device attributes, such as a special number of columns per line. When the console receives the question, it will respond by sending back the string listed next to “devatt” (short for “device attributes”). In this example, the console sends back the <nul> character, which is its way of saying “no, I don’t have any special attributes.”

The remote computer may also send the console an enquiry string that asks the console to send back an answer. The remote computer is asking the console to send back whatever string is listed next to “answer” (short for “answerback”) in the keyboard dictionary. It may be looking for specific information (e.g., what type of computer the console is), but there is no way the user can know exactly what the remote computer is asking.

When the console receives the enquiry, it will send back the answer string. In this example, the console tells the remote computer it is a VT100 terminal. The remote computer can then use that information to determine how it should “talk” to the console in a way the console can understand. What will happen if the remote computer is looking for a different answerback depends on how the remote computer is set up.

Although the user can change the strings associated with enhanced, devatt and answer, there usually is no need for any modifications and we strongly suggest that they be left the way they are.

×

Final Version

Final Version

Video Attributes

In this chapter, we'll discuss *video attributes*. Video attributes refer to special display characteristics applied to what appears on the screen. These attributes, in the case of a monochrome monitor, include underlining, boldface, blinking, reverse video, normal display, or certain combinations of these. In the case of a color monitor, these attributes include the color of the characters and the background upon which they appear, as well as whether or not they blink.

This chapter will cover the more detailed aspects of video attributes, including the method for altering attributes.

Video Attributes

We have provided the MONO.VID and COLOR.VID files with the console program so that under normal circumstances you won't need to make changes to the videomap dictionary. (A possible change would be to change the set-up from monochrome to color, which was discussed in the "Installing the Console Program" chapter.) However, it may be helpful for you to understand the contents of the videomap dictionary and how it is used by console. If you have a color monitor, you may even want to change the colors we have provided. To do that, it will be necessary to edit the videomap dictionary.

Console Only Uses Normal Text

Be aware that the console program ordinarily *only* uses normal text. Therefore, only the attribute(s) assigned to normal text will be seen on the console; the rest of the descriptions in VIDEOMAP.DCT will not be used by console.

Console will, however, honor any VT escape sequences that cause the text mode to change. So, it is possible for the console program to display text in different modes.

Issue the DOS **type** command to see the contents of the videomap dictionary.

A:\DCT>*type videomap.dct* **ENTER**

The codes used to define video attributes are found at the bottom of VIDEOMAP.DCT, so after issuing the command, let the text scroll to the bottom to see them.

Final Version

Monochrome

If you are using the default dictionary, MONO.VID, the videomap dictionary will look like this:

```
videomode    n      <7>
videomode    i      <15>
videomode    u      <1>
videomode    ui     <9>
videomode    b      <135>
videomode    bi     <143>
videomode    bu     <129>
videomode    bui    <137>
videomode    r      <112>
videomode    ri     <120>
videomode    ru     <112>
videomode    rui    <120>
videomode    rb     <240>
videomode    rbi    <248>
videomode    rbu    <240>
videomode    rbui   <248>

A:\DCT>_
```

Default (Monochrome) Videomap Dictionary Excerpt

The letters following the word “videomode” on each line describe the video attribute or combination of video attributes available for each character position on a VT100 or VT200 terminal screen.

Here is what each letter stands for:

Letter	Attribute
n	normal light on dark display
r	reverse video
u	underline
i	high intensity (or bold)
b	blinking

The attributes can be combined and applied to text that appears on the screen. Remember, however, that console only uses the normal attribute.

The attribute combinations listed in the dictionary describe the ways that characters can appear on a terminal screen. To the right of the letters in the videomap dictionary are the values that the console uses to determine the way text will appear on the PC screen under the same circumstances. For example, in the console videomap dictionary, the attribute combination “n” is assigned the value “<7>.” Therefore, any text that appears with the normal attribute will have a value of 7.

The values used are defined by IBM technical standards. When attributes are combined, the values in angled brackets are calculated by taking one value from each of the following three groups and adding them together.

Final Version

	Attribute	Attribute Value
Text:	No Display	0
	Normal	7
	Reverse	112
	Underlined	1
Blinking:	Non-blinking	0
	Blinking	128
Intensify:	Normal	0
	Intensify	8

Or, presented another way:

For text displayed as:	Start with attribute value of:	For blinking, add:	For intensification, add:
Normal	7	128	8
Reverse	112	128	8
Underlined	1	128	8

Note that certain combinations are not available on a monochrome monitor, reverse underline for instance. If you add the values for reverse and underlined together, you get 113. However, if you look at the screen display shown earlier, you will notice that the value for the reverse underlined attribute is defined as 112, which is the value for the reverse attribute alone. This tells you that the reverse underlined attribute is not available and that when that combination of attributes is used, the text will be in reverse video.

Since intensify and blinking cannot be displayed alone, normal text is assumed to be combined with them unless underline or reverse is specified. For example, in order to have normal text that has the intensify attribute, the value next to “n” in the dictionary should be 15. This is because the attribute value for intensify is 8 and the attribute value for normal is 7. Added together, they equal 15. The same holds true for blinking by itself: the value for blinking is 128. The value for normal is 7. Added, they yield 134.

After you edit the videomap dictionary and save the changes, it will be necessary to run the **makemux** program so that your changes will be installed.

A:\DCT>*makemux .. -i* **ENTER**

Now when you run console, the text will have whatever attribute you assigned to normal text.

Although you can come up with many interesting attribute combinations, normal text is usually the easiest and least distracting for people to read. For this reason, we suggest you don’t make any changes to the video attributes in VIDEOMAP.DCT.

Final Version

Color

If you are using the COLOR.VID dictionary, the file will look like this:

```
videomode    n      <31>            ; bright white on blue
videomode    i      <27>            ; bright cyan on blue
videomode    u      <30>            ; yellow on blue
videomode    ui     <29>            ; bright magenta on blue
videomode    b      <159>           ; blinking bright white on blue
videomode    bi     <155>           ; blinking bright cyan on blue
videomode    bu     <150>           ; blinking yellow on blue
videomode    bui    <149>           ; blinking bright magenta on blue
videomode    r      <112>           ; black on white
videomode    ri     <48>            ; black on cyan
videomode    ru     <96>            ; black on brown
videomode    rui    <80>            ; black on magenta
videomode    rb     <240>           ; blinking black on white
videomode    rbi    <176>           ; blinking black on cyan
videomode    rbu    <224>           ; blinking black on brown
videomode    rbui   <208>           ; blinking black on magenta

A:\DCT>_
```

Color Videomap Dictionary Excerpt

With this dictionary installed, different attributes for text are displayed as different colors rather than using the other attributes such as underline or reverse. For example, the “underline” attribute has been defined as yellow text on a blue background rather than appearing as underlined text on the screen. This color “highlighting” does not effect the way the text will be printed; it will still be printed as underlined text.

The reason color monitors don’t display underlined text with an underline is because they interpret the values in the videomap dictionary differently than do monochrome monitors. In other words, the attribute values correspond to a completely different attribute set depending on the type of monitor you are using. A value of 1 on a monochrome monitor, for instance, corresponds to the underline attribute. On a color monitor, a value of 1 would correspond to blue letters appearing on a black background.

Below are the attributes available on a color monitor and their corresponding attribute values. You can choose one attribute value from each of the three groups below, add them together, and insert the total into the videomap dictionary next to the “n” to define how normal text will appear.

Final Version

Foreground Color Attribute

Attribute	Attribute Value
Black	0
Blue1	
Green	2
Cyan	3
Red	4
Magenta	5
Brown	6
White	7
Gray	8
Bright Blue	9
Bright Green	10
Bright Cyan	11
Bright Red	12
Bright Magenta	13
Yellow	14
Bright White	15

Background Color Attribute

Attribute	Attribute Value
Black	0
Blue	16
Green	32
Cyan	48
Red 64	
Magenta	80
Brown	96
Light Gray	112

Blinking Attribute

Attribute	Attribute Value
Non-blinking Characters	0
Blinking Characters	128

If you don’t like the way normal text looks, you can edit the dictionary with whatever values correspond to the color you want normal text to be. To do this, you need to calculate the value of the new attribute you want and then use a DOS editor such as **edlin**.

To calculate the value associated with the color attribute combination you desire, choose one item from each of the three available attribute groups. That is, choose the foreground color you desire, the background color you desire, and whether or not you want to apply

Final Version

the blinking attribute as well. Add the individual attribute values together, and the result will be the value you enter in the dictionary.

For example, normal text may appear a little too intense for your taste on your color monitor. The color videomap dictionary assigns the attribute value of 31 to normal, “n,” which is a combination bright white on a blue background: 15+16=31. You may want to change the bright white foreground attribute to white for normal text. To keep a blue background, add the attribute values for white foreground and blue background. You won’t be adding the blinking attribute in this case, so the value from that group will be zero:

Foreground white	7
Background blue	16
Blinking characters	0
Total	23

It is this value, 23, which you will place between angle brackets in the videomap dictionary next to “n.”

After you edit the videomap dictionary and save the changes, it will be necessary to run the **makemux** program so that your changes will be installed.

A:\DCT>*makemux .. -i* **ENTER**

The next time you run console, normal text will appear as white over a blue background instead of as the bright white that had appeared previously. The blue background color that is assigned to “n” controls the color that is shown whenever the screen is cleared. That is, the screen background color will be whatever background value is included in the total value assigned to “n.”

×

Final Version

Character Mapping

Translating MCS and IBM Characters

The characters that can appear on your console screen are not limited to just those which can be found on the keys. There is a set of characters beyond the standard keyboard characters that are part of what is called the *IBM Extended Character Set*. These are an extension to the standard ASCII character set and include characters such as å, ¥, ¿, and æ. While standard ASCII character values run from 0 to 127, IBM extended characters' corresponding values run from 128 to 254.

There are also other character sets, used by different computer systems, to provide these extended characters (which are sometimes called *international* or *compose characters*). In addition to the IBM Extended Character Set, there is also the DEC VT200 Multinational Character Set. It is this character set that is used internally by the console program. Since the console software runs on a PC, a facility is needed to translate DEC values to IBM character values, and vice-versa.

Translations of this kind take place under the following conditions:

- When a character is sent to be displayed on the console screen
- When a character is sent to a printer connected to the console
- When a character is sent to the log file

For example, the value that generates a ¥ symbol on a terminal may not generate the same symbol on the console screen. A translation is needed so that the correct character will be displayed in both places.

In this chapter, we will discuss the portion of the videomap that handles such translations, and the conditions when these translations are invoked. For information about generating international from the console, check the "Customizing Your Keyboard" chapter as well as Appendix A. Also refer to the manuals that came with your PC for more information about how to generate international characters on your PC.

IBM to MCS Character Mapping

The character set translations take place in the videomap dictionary. Issue the DOS **type** command to see the VIDEOMAP.DCT file:

A:\DCT>*type videomap.dct* **ENTER**

The first section that scrolls past will look like this:

Final Version

Videomap Dictionary Excerpt

This section of the dictionary contains entries with “ibmtomcs” in the first column. MCS (**M**ultinational **C**haracter **S**et) refers to what we call MCS characters. The second column contains the IBM character or character code being translated, and the third column contains the MCS multinational character or c being used as the description. Some of the translations are identical, (letters, for example) and others include very different translations. In the case of differing translations, IBM characters are represented in the dictionary by numbers between angled brackets, and MCS characters are represented by character sequences that help clarify the international character being defined.

ibmtomcs Translations are Bi-Directional

Even though the first section of the videomap dictionary says “ibmtomcs,” translations from MCS to IBM are also handled here. You can think of the dictionary’s first section, then, as a bi-directional set of translations.

MCS character sequences such as “_i^” are used here instead of their numerical equivalent to give you an idea of the MCS character involved. For example, the sequence “_i^” generates an î character. You could put the MCS value <206> here instead of the character sequence, but if we used only numerical values in the dictionary, you would have to refer to the chart at the back of this chapter to understand which MCS character was involved in each translation.

Let’s look at the entry that contains “ibmtomcs <135> _c,”. The numerical value is the IBM code for a c-cedilla (ç). When the videomap dictionary is interpreted by the **makemux** program, the sequence “_c,” is matched to the MCS code for a c-cedilla. This

Final Version

entry in the dictionary, therefore, effects a match between IBM code 135 and MCS code 231. Check the chart in the reference section to see how we came up with this correspondence.

The MCS to IBM translation is discussed next.

MCS to IBM Character Mapping

Think of MCS as the language that the console understands and IBM as the language that your PC screen understands. Console recognizes only MCS characters. The console software will need to translate MCS characters to IBM characters when it sends the characters to an output device (a disk, printer, or screen) because that is the way the output device reads characters.

For example, the entry in the videomap dictionary that matches the MCS character sequence "_!!" (MCS value 161) with the IBM value 173 (the IBM value for an inverted exclamation point) ensures that a log file (or any file viewed with the console **view** command) containing the MCS code 161 will appear as a ¡ on the PC screen.

There are some characters that are available in MCS that are not available in IBM. These include some upper case characters like È, Ò, and Â. Whenever a MCS code that specifies one of these characters is sent to the PC screen or printer, it is translated into the IBM code that corresponds to the most similar character available, usually the lower case of the same character. The MCS code that generates the character Â, for example, is translated to the IBM code that generates an â. These translations are specified in the second section of the videomap dictionary, where the entries are preceded by "mcstoibm."

Final Version

```
ibmtomcs      <174>  _<<<<
ibmtomcs      <175>  _>>
ibmtomcs      <225>  _ss
ibmtomcs      <230>  _u/
ibmtomcs      <237>  _o/
ibmtomcs      <241>  _-+
ibmtomcs      <248>  _o<SP>
ibmtomcs      <253>  _2^

mcstoibm      _A^    <131>
mcstoibm      _A`    <133>
mcstoibm      _E^    <136>
mcstoibm      _E"    <137>
mcstoibm      _E`    <138>
mcstoibm      _I"    <139>
mcstoibm      _I^    <140>
mcstoibm      _I`    <141>
mcstoibm      _O^    <147>
mcstoibm      _O`    <149>
mcstoibm      _U^    <150>
mcstoibm      _U`    <151>
mcstoibm      _Y"    <152>
mcstoibm      _A'    <160>
mcstoibm      _I'    <161>
mcstoibm      _O'    <162>
mcstoibm      _U'    <163>
mcstoibm      _O/    <237>
```

Videomap Dictionary Excerpt

Note that since MCS is a superset of IBM (that is, it contains everything the IBM Extended Character Set contains, and then some), these translations only apply when one of these upper case MCS characters needs to be translated to a similar IBM character; the translation does not work in reverse since all IBM characters are mapped to their MCS equivalent in the first section of the videomap dictionary.

The MCS characters that are not included in the videomap dictionary are defined automatically as “null.” These include such non-displayable characters as <ESC> and <HT> (which are, respectively, an escape command and a horizontal tab).

×

Final Version

Section 2
Usage

Final Version

Final Version

The Console

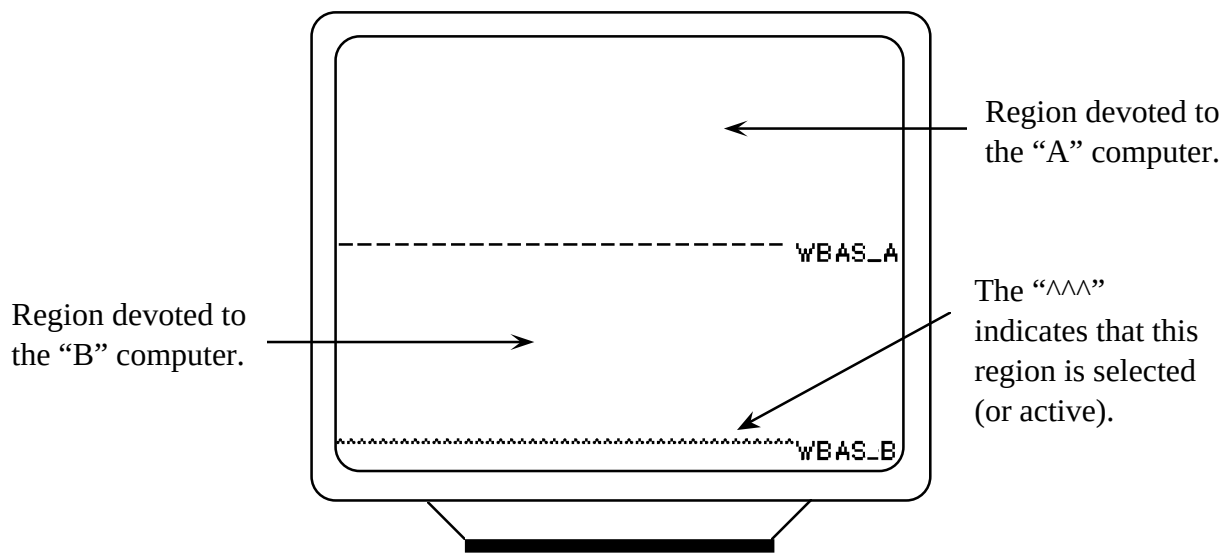
Monitoring and Controlling Your System

The Console Multiplexor (“console”) program is a DOS-based program that runs your console. It is from the console that the System Manager(s) communicate with the host computers. The console program allows System Managers to send commands to the computers to perform maintenance and to set options for the system and users.

This chapter will introduce you to the console, so you’ll feel comfortable using it to perform the procedures described in this section and in the News Operations Manual. Since one of the console’s primary functions is to display the output from each computer in your system, we’ll begin by describing how your console divides its screen between the computers.

The Console Screen

Though it controls multiple computers, your console only has one screen, which it must divide into regions to separate the output of each computer. For example, the picture below shows a console in a dual-computer system, so the console screen is divided into two regions. (Your console screen will have as many regions as there are computers in your system.)


Region devoted to the “A” computer.
WBAS_A
Region devoted to the “B” computer.
The “^^^” indicates that this region is selected (or active).
WBAS_B

Console Screen for a Dual-Computer System

To identify which region belongs to which computer, the console displays the name of the computer that a region represents in that region’s lower right corner. In the example above, the top region displays the output from the “A” computer, and the bottom region displays the output from the “B” computer.

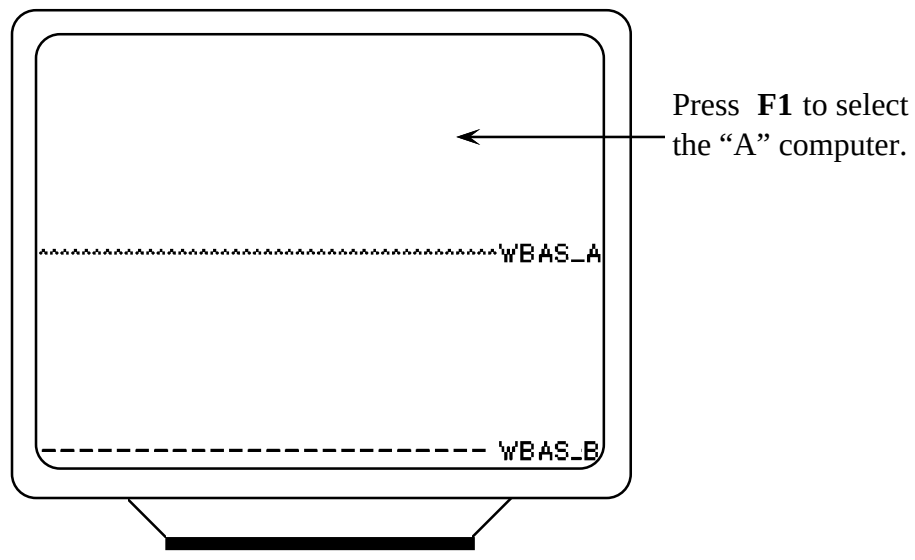
Part of the console’s function is to receive and display messages from the computers. The other half of the console’s job, of course, is to let you send commands to your

Final Version

system’s computers. Before you can send a command from your console, however, you must tell the console which of your system’s computers you want to receive that command.

To indicate which computer(s) you want a command sent to, you must first *select* the computer(s). When you select a computer, the bottom line of that computer’s region will change from a row of dashes to a row of “^” characters. In the diagram above, the “B” computer is currently selected.

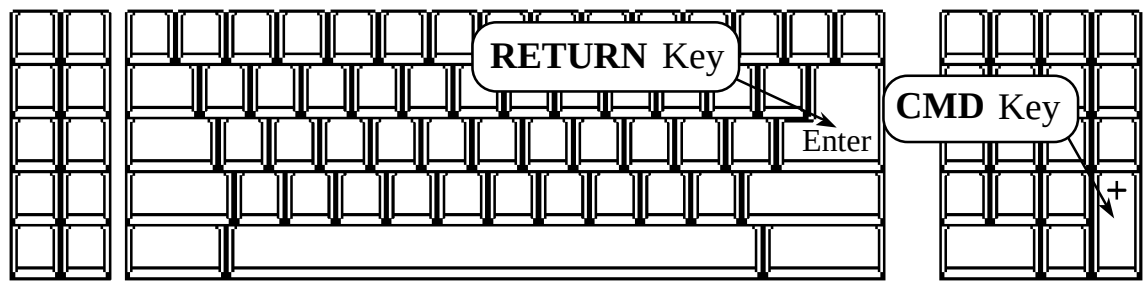
Selecting a computer is easy, since the commands to do this are usually assigned to several of your console keyboard’s function keys. On most systems, pressing the **F1** key will select the “A” computer and pressing **F2** will select the “B” computer. Try it yourself—if the “B” computer is currently selected, press **F1** to select the “A” computer. (If your keyboard function keys are not set up, we will tell you how to manually enter commands as well as how to assign commands to function keys a little later in this chapter.)


Press F1 to select
the “A” computer.
^^^

The commands assigned to these function keys are called *console control commands*, since they control the operation of the console. In a moment, we'll show you how to enter console control commands manually, rather than relying on the function keys. First, though, you need to know how to tell the console that the command you are about to enter is intended for the console and not one of the computers.

The CMD Key

As shown below, your console's keyboard is just like a standard PC keyboard. The only difference is that the "+" key (labeled "Enter" on some keyboards) in the lower right corner of the keyboard has a special function. This is your console's **CMD** key. It allows you to toggle between sending commands to your system's computers and sending commands to the console itself.



Typical Console Keyboard

When you press the **CMD** key, the "COMMAND" prompt will appear at the bottom of the currently selected region(s). Your cursor will move to this prompt so that you can enter console control commands.

Where's the RETURN Key?

You must press **RETURN** after entering a console control command to execute that command. As shown above, the **RETURN** key is frequently labeled "Enter." On enhanced keyboards, however, this key is labeled "Return." If you find this confusing, just remember that on all keyboards, the **RETURN** key is always in nearly the same position as the carriage return on a standard typewriter keyboard.

Now that you know how to send commands to the console, let's take a look at the console control commands you can use to select any computer or group of computers in your system.

Selecting a Computer

To select a computer, use the **computer** command followed by the names of the computers you want to select. Since **computer** is a console control command, begin by pressing **CMD** to bring up the "COMMAND" prompt. Then type "c" (short for "computer"), followed by a space and the computers you want to select. For example, to select the "A" computer, type in the following and press **RETURN**.

```
COMMAND c a
```


Final Version

You could select all the computers in your system by following the **computer** command with the names of your system's computers. An easier way, however, is to follow the **computer** command with an asterisk as shown below. The console will interpret the asterisk to mean all the computers in your system.

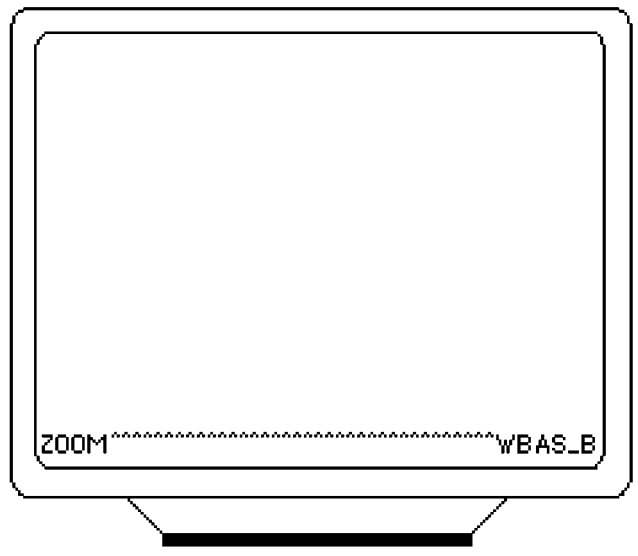
```
COMMAND c *
```

If You Mistype a Console Control Command

If you mistype a command but haven't pressed **RETURN**, you can start over by pressing **CMD** once to clear the line and a second time to bring the "COMMAND" prompt back. Alternatively, you can use the **BACKSPACE** key to move the cursor back and type over the error.

Zoom

In addition to the **computer** command, you can also use the **zoom** command to select a computer. Unlike the **computer** command, however, **zoom** selects just one computer at a time and devotes the entire console screen to that computer. The diagram below, for example, shows the console screen after "zooming in" on the "B" computer's region.


ZOOM^^^

Which Computer to Select?

By now, you've probably begun to wonder how you know when to select one computer and when to select all computers. The answer, as you might suspect, depends on what you are doing.

With some commands, it doesn't matter which computer you select. With others, however, it's crucial that you select one particular computer. With still other commands you must select all computers. To help you make sure you always have the right computer(s) selected, every procedure in this manual and the News Operation Manual indicates which computer(s) you should select before executing a particular command.

Also, to give you an idea of what kinds of activities require that you select a particular computer, all computers, or just any computer, we've included the following guidelines:

Select the Master Computer When...

You must select the master computer when you make major changes to the database (e.g., when you purge stories, build up the free list, or examine the database for errors).

Select Any Computer When...

You can select any computer when you make minor changes to the database. Minor changes include: modifying user and directory traits; backing up and restoring the database; listing user, directory, or configuration information; sending messages; removing edit locks; and checking the amount of free space available.

Select All Computers When...

You must select all computers when you are going to modify site files or shut down the system.

Select a Specific Computer When...

You must select a specific computer when you are going to execute a command that affects a device connected to that computer. For example, before you can restart a CCU on the "A" computer, you must first select that computer.

Remember, though, that these are only guidelines; double-check the specific instructions.

Console History

Along with connecting you to the computers in your system, the console can also record everything it has displayed in the last few days—including commands entered at the console and error messages. This record is called the *console history*, or simply *history*. As you can imagine, this audit trail of console activity is an invaluable diagnostic tool if a problem occurs.

Final Version

Recent history (approximately the last 24 hours, depending upon console activity) is held in the console's memory. If you wish, you can also have the console store older portions of history on a floppy diskette. This is called *disk logging* and is usually enabled when your system is installed. Disk logging allows the console to record up to 64 kilobytes—several day's worth—of history on a floppy diskette.

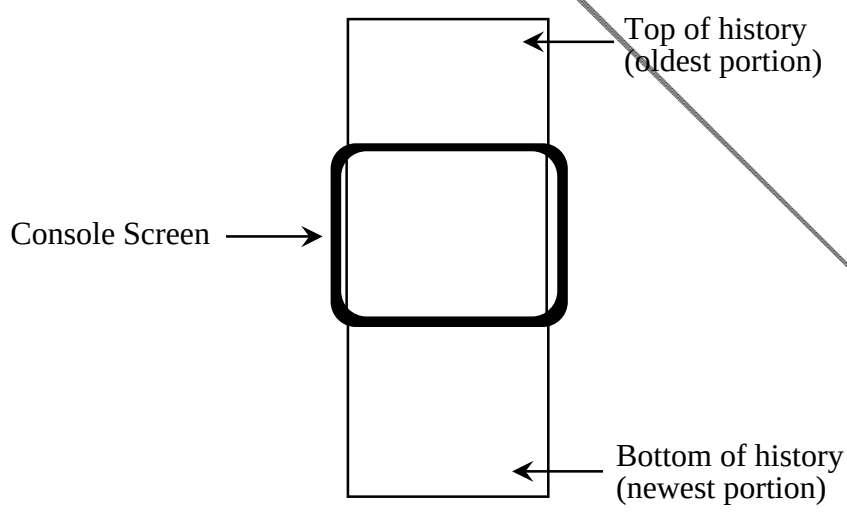
The "LOG" Message

When the console periodically saves a portion of history to a disk file, it will briefly display "LOG" at the bottom of the region belonging to the computer whose history it is saving.

In the pages that follow, we will first show you how to view recent history (the portion held in the console's memory). Then we'll show you how to take a look at older portions of history that have been written to disk.

Viewing Recent History

To understand the commands you use to view recent history, it's helpful to think of recent history as a long sheet of paper over which your console screen moves up and down. As shown below, the oldest portion of recent history is at the top of the sheet of paper and the newest portion is at the bottom.



Commands that move you up towards the top of this imaginary sheet of paper take you backwards through history, and commands that move you down take you forwards through history. These commands, as well as two that allow you to print portions of the history, are described in detail below.

Top

This command will move you to the oldest line in the history.

Bottom

This command will move you to the newest line in the history.

Final Version

Up [<number of lines>] or [<keyword>]

The **up** command moves you up some number of lines in the history. If you follow **up** with a number, it will move you that many lines back in history. For example, “up 30” will move you back 30 lines. If you do not follow **up** with a number, it will move you back just one line.

If you follow **up** with a word, it will search backwards through history from your current position for that word. For example, to search backward for a line containing “dbpurge,” type “up dbpurge” and press **RETURN**.

Up is not case sensitive, so if you ask it to find “DBPURGE,” it will consider “dbpurge” to be a valid match. If **up** doesn’t find the keyword before reaching the top of history, it will beep and return to the current line. If you follow **up** with a number greater than the number of lines between your current position and the first line in the history, **up** will move you to the first line in the history.

Down [<number of lines>] or [<keyword>]

This command moves you down (forward) some number of lines in history. If you follow **down** with a number, it will move you that many lines forward in history. For example, “down 10” will take you 10 lines forward. If you do not follow **down** with a number, it will move you down just one line.

Following **down** with a word will cause it to search forward from your current position for that word. For example, to move forward to a line containing “list,” type “down list” and press **RETURN**.

Down is not case sensitive. If you ask it to find “LIST,” **down** will consider “list” to be a valid match. If **down** hasn’t found the keyword by the time it reaches the bottom of history, it will beep and return to the current line. If you follow **down** with a number greater than the number of lines between your current position and the last line in the history, **down** will move you to the last line in the history.

List [<number of lines>] or [all]

The **list** command allows you to send a specified number of lines of history to the printer that is connected to the console. If you follow **list** with a number, it will print that many lines of the history beginning at the current line. If you follow **list** with the word “all,” **list** will print everything from the current line to the newest line. To print the current line, just type “list.”

When you use **list**, “PRINTER” replaces the “HISTORY” prompt. If “PRINTER” is displayed but nothing is being printed, check to make sure that the printer is plugged in and turned on. Also, make sure that the printer is on-line and not out of paper.

If you want to abort a **list** command while it is printing, just press any key. The printing will stop soon thereafter, and your position in history will move to the last line sent to the printer.

Final Version

Print [<number of lines>] or [all]

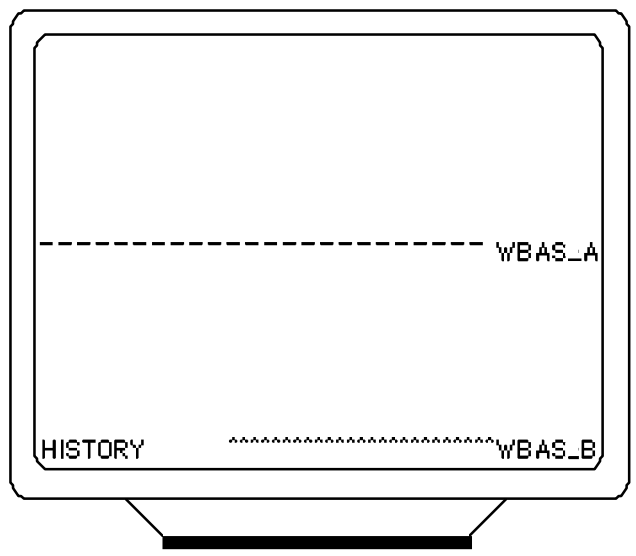
The **print** command prints a number of lines of history on the console screen. If you follow "print" with a number, the console will display that many lines of history beginning at the current line. Following **print** with the word "all," on the other hand, will display everything from the current line to the newest line. If you do not follow **print** with anything, it will display only the current line.

If you want to abort a **print** command while it is printing, simply press any key. The console will stop at the line last printed on the screen, and your position in history will move to that line.

Sample Session

Since the console records history for each computer in the system, you can choose to look at just one computer's history or all computers' histories. To do this, select the computer(s) whose history you want to examine, before executing a history command.

After selecting the computer(s) whose history you want to look at, press **CMD** and type in the history command you want to use. For most commands, you can use just the first letter of the command (e.g., "u" for "up"). When you press **RETURN**, the console will execute the command and replace the "COMMAND" prompt with a "HISTORY" prompt to indicate that you are viewing history. Note that the histories of all selected computers will be displayed simultaneously.



Examining the "B" Computer's Recent History

While you are viewing history, the computer(s) whose history you are viewing will not display any messages on the screen. When you have finished viewing history, press **CMD** to clear the "HISTORY" prompt from the screen. Once this prompt has been cleared, the computers will be able to begin sending messages to the console again.

Final Version

Exit From History When You Are Done

It is important that you remember to exit from history by pressing the **CMD** key when you are finished viewing history. If you leave the prompt up for a long time (several hours), the computers whose histories you are viewing may shut down.

Reading Older History

Older portions of history are held in files on a diskette in one of your console's disk drives. Usually, when a system is installed, the console will have been set up so these files are on the diskette in drive B. (If your PC has a hard drive, disk logging will probably be set up so the files are saved on a diskette in drive A.)

Disk Logging Must Be On

Disk logging must be enabled in the console's *configuration file* (described in "Console Configuration File" in this section), or these files will not exist. Note that the names of these files are set in the console configuration file as well.

If disk logging is enabled, the console will save the older portion of each computer's history in four separate files on the floppy diskette. For example, a dual computer system might have the older portions of the histories for the two computers saved in the files shown in the table below:

Computer	History File	Portion of History in The File
"A" Computer	LOG.A1	The latest portion
	LOG.A2	A slightly older portion
	LOG.A3	An even older portion
	LOG.A4	The oldest portion
"B" Computer	LOG.B1	The latest portion
	LOG.B2	A slightly older portion
	LOG.B3	An even older portion
	LOG.B4	The oldest portion

Console History File Chart

As you can see, all the files have names beginning with "LOG." Each file however, has a different *extension* (the portion of the name separated from "LOG" by a period). Each extension begins with a letter that identifies which computer's history the file contains. For example, the "A" computer's history files all have extensions that begin with the letter "A."

Each extension also has a number that indicates which portion of a computer's history is held in that file. For example, LOG.A4 holds the oldest portion of the "A" computer's history, while LOG.A1 holds the most recent portion of the history that has been saved on disk for that computer.

Final Version

In this example, when the console saves the “A” computer’s history, it will save the information in the LOG.A1 file. When LOG.A1 is full, the history information will get “pushed” to (i.e., saved in) the LOG.A2 file. Each time the “A” computer’s history is saved, information will be moved to the next file (LOG.A1 to LOG.A2, LOG.A2 to LOG.A3 and LOG.A3 to LOG.A4).

When all the log files are full, information will be “pushed off” LOG.A4. That history information will be lost. This way you’ll always have a few day’s worth of history to check if there is a problem. (You shouldn’t need any more history than what’s in the console’s memory and in the four log disks.)

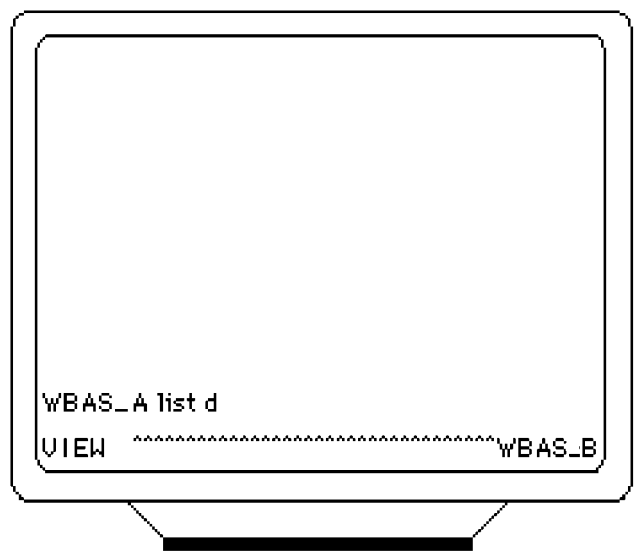
To look at the history held in these files, you’ll need to use the **view** command. In order to use this command, you must first make sure that only one region is displayed. So before using **view**, use **zoom** to display just one region—it doesn’t matter which one.

After zooming in on one computer region, press the **CMD** key to bring up the “COMMAND” prompt. Then type “v” (short for “view”) and the name of the file you want to view. Since the history files are usually on the B drive, you must precede the file name with “b:” to indicate that you want to view a file on the B drive.

For example, to view the most recent portion of the “A” computer’s history that has been saved to disk, bring up the “COMMAND” prompt and use **view** to read the LOG.A1 file as shown below.

```
COMMAND v b:log.a1
```

When you press **RETURN**, the “COMMAND” prompt will change to “VIEW” to indicate that you are viewing a file, and the first line in the file will be displayed at the bottom of the console screen. To scroll through the file, press **RETURN**. Each time you press **RETURN**, the file will scroll up one line.



Reading an Old History File

Final Version

While using **view**, you can only move down through the file. When you reach the end of the file, the console will beep, indicating that there is nothing more to display. When you are done viewing the history file, press **CMD** to return to the regular console screen. Since what you are viewing is already a part of history, it will not be recorded into history a second time as you view it.

Logclose

Any file with "1" in its extension will contain the most recent portion of history that has been written to disk. If you would like everything in recent history to be written to this file before viewing it, use the **logclose** command as shown below:

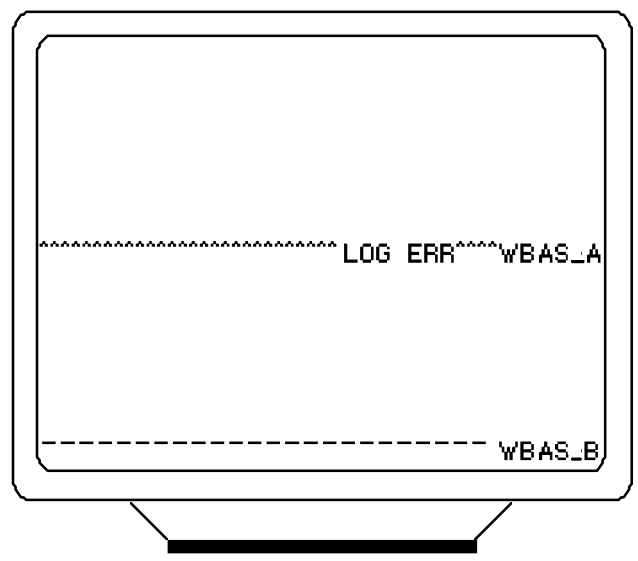
```
COMMAND logclose
```

This will write everything in recent history for the selected computer(s) to disk. It also turns disk logging off, so you should follow it with a **logopen** command as shown below to turn disk logging on again.

```
COMMAND logopen
```

Log Error

If the console software detects a disk error while writing a portion of history to disk, it will display the "LOG ERR" message shown below. This error may be due to a missing or improperly formatted diskette or a physical defect on the diskette. Whatever the cause, the console will suspend disk logging until you fix the problem.



"LOG ERR" Message

If you see this message, first check to make sure that the diskette is inserted in the correct disk drive (usually drive B) and that the drive handle is in the down (closed) position. If the diskette is properly inserted, it may not have been formatted correctly, and you should reformat the diskette as described in your PC manual.

Final Version

If, when you attempt to reformat the diskette, the format fails and the PC displays an error message, then the diskette is probably defective, and you will need to replace the diskette with a new, properly formatted one.

Once you've fixed the problem, you'll need to restart disk logging using the **logopen** command. Press **CMD** to bring up the "COMMAND" prompt and type "logopen" as shown below:

```
COMMAND logopen
```

When you press **RETURN**, the "LOG ERR" message will disappear and disk logging will resume.

Changing Log Disks

If you want to replace the diskette containing your history files, you should make sure that the console doesn't try to save a portion of history to diskette while you are making the change. To do this, temporarily disable disk logging using the **logclose** command. Press the **CMD** key to bring up the "COMMAND" prompt, type "logclose," and press **RETURN** as shown below:

```
COMMAND logclose
```

This will disable disk logging, so nothing in history will be saved to disk until you enable disk logging again. After placing the new diskette in the proper drive and closing the drive handle, use the **logopen** command to resume disk logging.

Console Commands

So far, we've only discussed console control commands. As we mentioned earlier, though, you will also use commands that control the computers in your system. These commands are called *console commands* to distinguish them from terminal commands and console control commands.

As you read this manual and the News Operations Manual, you will be introduced to the console commands that you will need to use as a System Manager. For now, keep in mind that, like all other commands, these will not execute until you press **RETURN**. Also, keep in mind that console commands *must* be typed in lower case.

If you make a mistake when typing a console command, use **BACKSPACE** to move the cursor back over the mistyped portion and then type over it. Alternatively, you can abandon the entire line and start over by pressing the **@** key.

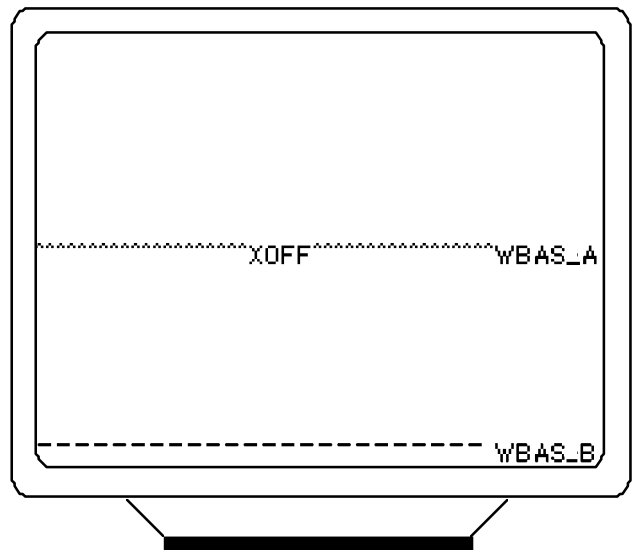
Final Version

Aborting a Console Command

If, after executing a console command, you decide that you need to stop it, press the **DELETE** key or the **ALT** key. If that doesn't work, press the "\" ("backslash") key while holding the **CTRL** key down. This will stop the command and display the "login:" prompt. Type "so" (short for "system operator" at the "login" prompt and press **RETURN** to log in.

Pausing the Screen Display

Often, when the output from a console or history command is longer than one screen, you'll want to pause the display so that you can look at one portion before the next portion is displayed. You can do this by pressing **^S** (the **CTRL** key and the **S** key simultaneously), which temporarily stops the console screen from scrolling. When you stop scrolling, "XOFF" will be displayed at the bottom of the selected region.



"XOFF" Message

When you are ready to resume scrolling, press **^Q**. By the way, it's important that you don't pause the screen display for too long. If you do not resume scrolling within sixty seconds, the console will resume scrolling for you. You can press **^S** again if you want to temporarily stop scrolling once more. Note that pressing **^S** does not have any effect on a remote console.

Function Keys

Earlier, we showed you how to use three of your console's function keys to select different computers. After you've used the console for a while, you'll notice that there are other commands that you use often. You can save time by assigning these commands to the function keys on the console keyboard.

If you have an enhanced keyboard, you can assign definitions for up to 14 function keys: **F1 - F12**, **+** and **ENTER**. (Only enhanced keyboards have **+** and **ENTER** keys; so if your keyboard has them, you have an enhanced keyboard.) Note that if the **+** and **ENTER** keys do not have commands assigned to them, they will behave as a **CMD** key.

Final Version

When you assign a definition to a function key, that definition is recorded in a file called CONSOLE.KEY in drive A. This way, the definitions assigned to your console's function keys will remain, even if you turn the console's power off.

Assigning a Definition

To assign a command to a function key, press **CMD** to bring up the "COMMAND" prompt and then type the name of the function key you want to use (e.g., "f5"). Follow that with "=" and the command you want to assign to the function key. The general format for doing this is shown below:

COMMAND <function key>=<command>

As an example, let's say that you frequently use the **list t** command to find out who is logged in on the system. To assign this command to **F9**, you would press **CMD** and type the following:

```
COMMAND f9=list t
```

When you press **RETURN**, "list t" will be assigned to **F9**. Thereafter, pressing **F9** will cause "list t" to be displayed on the screen, and all you'll need to do is press **RETURN** to execute the command.

To include **CMD** or **RETURN** in a definition, use "{" to represent **CMD** and "}" to represent **RETURN**. For example, suppose you want to create a function key that would select all computers. As you saw earlier, this requires that you press **CMD**, type "c *" and press **RETURN**. To assign this sequence of keystrokes to **F8**, for example, bring up the "COMMAND" prompt and type in the following:

```
COMMAND f8={c *}
```

Next, press **RETURN** to assign that definition to **F8**. Now, you can use **F8** to select all computers.

Changing a Definition

To change a function key's definition, you simply assign the new definition to the key. For example, suppose you wanted to change the definition given to **F9** in the earlier example so that it would execute the **list t** command. To do this, you would assign this command as you did before, only this time you would add a "}" to the end, as shown below:

```
COMMAND f9=list t}
```

Now, pressing **F9** will not only display "list t," but it will execute it as well.

Note that if you view CONSOLE.KEY after changing a function key's definition, the file will not show the change. You must exit the console program and then go back into it to see the changed definition. However, the key will perform the new command.

Final Version

If You Forget a Definition...

You should label your function keys to identify the definition assigned to each. However, labels sometimes fall off or become smeared. If you forget what has been assigned to a function key, you can display the key's definition on the console screen.

To display a key's definition, press **CMD**, type in the name of the function key and press **RETURN**. The console will display the definition assigned to that key. For example, if you wanted to find out what had been assigned to **F9**, you would press **CMD**, type in "f9" and press **RETURN**. The command line would look like the one shown below:

```
COMMAND f9 = list t}
```

After examining the definition, press **CMD** a second time to clear this display from the screen.

Remote Console

As we mentioned at the beginning of this chapter, the console software also provides for a modem to be connected to it. This allows someone in another location to use a computer and a modem to call up the console software and log in, effectively allowing the person to turn his or her computer into a "remote console."

The primary use for a remote console is to allow technicians and System Managers who are not in the computer room to perform diagnostic and maintenance work on your system. Also, as you'll see in "Sample Connect Services" in Volume 2 of the News Operations Manual, the modem can be used to allow the System Manager to do maintenance and diagnostic work directly from his or her News terminal.

Here, we'll discuss how to dial in over a modem line, what you should expect to see on the remote screen and how to execute commands remotely.

Dialing In

To dial in to the console, you must have a terminal or a computer that transmits and receives ASCII characters. You must also be familiar enough with your modem and communications software to be able to set it to eight data bits, no parity, one stop and one start bit, Xon/Xoff handshaking, and the correct baud rate. (The baud rate is set to either 1200 or 2400 baud in the console's configuration file. See "Console Configuration File" in this section for details.)

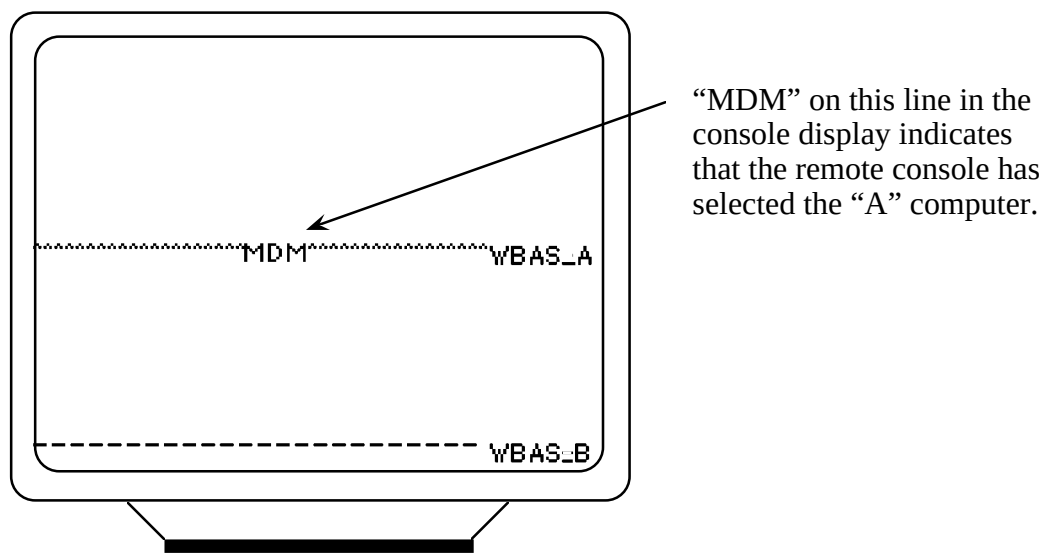
To prevent unauthorized people from dialing in to the console, access is protected with a password. (See "Console Configuration File" for details on setting the password.) When you dial in, you'll see the "PASSWORD" prompt on your screen. If you don't see this prompt immediately, pressing **RETURN** should display it. After you enter the correct password, the console will connect you to the first computer listed in the console configuration file, usually the "A" computer.

Final Version

If you know someone is going to need to log in to your console (e.g., a DEC service technician) and you don't want to tell that person your modem password, you can set it up so he or she can log in without it. The **modem open** console command ("m o" for short) allows anyone who logs in to your system to have access to your console, without requiring him or her to know the password. Because **modem open**, which can only be entered at the console, circumvents your security, you should only enter the command right before the "guest" user logs in. Be aware, though, that if anyone else logs in before the guest user, that person will have access to your console.

After the guest user has logged out of the console, be sure to enter the **modem** command ("m" for short) to reactivate the password. (Don't enter the command until the user has logged out, or you will log the guest out.) Hereafter, anyone who logs in will be required to know the password before gaining access to the console. If you forget to enter the **modem** command, anyone who logs in to your system through a modem will be able to connect to your console without being asked for the password.

At the console, "MDM" is displayed at the bottom of the region representing the computer that was selected from the remote console (see below). Characters typed at the remote console are sent to that computer and displayed on its console region. Likewise, characters sent by that computer are displayed both on the console and the remote console.



Console During Modem Connection

Executing Commands Remotely

Once logged in, you can enter console commands and review history almost as if you were seated at the console. Just as at the console, these commands may be abbreviated using the first letter in each command. All commands except **zoom**, **escape** (a dialog command that is explained in Volume 2 of the News Operations Manual), **logopen**, **logclose** and the history command **list** are available from the remote console. (Note that if you try to execute **logopen**, **logclose** or **list** from a remote console, you will logout the modem connection.)

There are also some differences when using a remote console. The biggest of these concerns the remote console's screen display. Because modems transfer information

Final Version

relatively slowly, they can't send information fast enough for the remote to be able to display a separate region for each computer as the console does. Instead a remote console can only show the output from the selected computer.

If you select more than one computer, the remote console will only display the screen input and output for one computer. What you type will be sent to all the selected computers, but you'll only see output from the first computer in the list.

When you select two or more computers, the order in which you list the computers in the **computer** command determines which computer's display you'll see. For example, if you used "c ba" to select computers "B" and "A," you would only see the output from the "B" computer on the remote console, even though what you type will be sent to both "A" and "B."

The "COMMAND" prompt is also different. It is displayed as "C-A?" on the remote console. The "C" is equivalent to "COMMAND" and the "-A" is included in the prompt to indicate which computer is selected. For example, if you selected computers "A" and "B," you would see the prompt "C-AB?".

The history prompt differs in the same way. For example, if you enter a history command while computers "A" and "B" are selected, the prompt will be "H-AB?" indicating that you are viewing the history for computers "A" and "B."

Similarly, when viewing a DOS file on the A drive the prompt will be "V-A?" on the remote console. Note that only one computer can be selected when issuing the **view** command from a modem.

Another difference concerns the keyboard. Since a wide variety of devices can be used to dial in, your remote console may not have a **CMD** key. If not, you can use the **ESC** key or type **^[** (hold down **CTRL** and press "[") when you need to press the **CMD** key.

Remote Console Function Keys
If the device you are using as a remote console has function keys, please note that the definitions assigned to the console's function keys are *not* assigned to the remote console when you log in. You will have to manually enter the commands.

Logging Out

When you have finished using the remote console, you must log out by pressing **ESC**, typing "l" (for "logout"), and pressing **RETURN**. When you do this, the remote will redisplay the "PASSWORD" prompt, and you should hang up your modem. At the console, "MDM" will disappear from the screen.

If you are at the console and discover that someone had been using the remote console but did not log out when he or she was done, you can log the modem out yourself. To do this, press **CMD**, type "m" (for "modem"), and press **RETURN**.

Final Version

You can set up your system so that it will automatically log out a user who has disconnected the modem without logging out or has left the connection inactive for a specified period of time. See a description of the “timeout” keyword in “Console Configuration Keywords” later in this section.

Disconnecting the Modem

If you need to disconnect the modem and there is a remote console logged into your console, log out the remote console (using the **modem** command described above) before disconnecting the modem from the console.

If You Can’t Get a Prompt on the Console...

If the computers are not responding to commands and are not displaying messages, first check to make sure that someone has not stopped scrolling by pressing **^S**. If they have, “XOFF” will be displayed under each computer region where scrolling has stopped. Press **^Q** to start scrolling again.

If you do not see an “XOFF” message, then the problem may be that the computer ports have stopped sending and receiving. To start them sending and receiving again, press **CMD**, type “x” and press **RETURN**.

If this does not restore the console, the problem may be with the console’s I/O ports. Static electricity may have “frozen” one or more of these ports. You may be able to unfreeze these ports using the **reset** command. To do this, select the affected computer(s), press the **CMD** key and type “r” (for “reset”) as shown below:

```
COMMAND r
```

Another potential cause of a frozen console is a program that won’t stop. If you suspect that this is the problem, select the affected computer(s) and press ^ \. This will stop the program on those computers and cause them to display the “login:” prompt. When you see this prompt, type “so” and press **RETURN** to log in as system operator.

If the console still does not respond, exit the console program and restart it as described in the following pages.

Exiting the Console

You may need to exit the console program if the console has been exposed to static electricity. A jolt of static electricity can garble the console, causing it to stop. You’ll know this has happened if the console screen freezes so that no new messages are displayed and no commands are accepted.

If you are not able to unfreeze the console screen using **reset** or by pressing “\” while holding down the **CTRL** key as described above, you’ll need to exit the console program. Then you can load a new uncorrupted copy of the console software into memory by restarting the console.

Final Version

To exit the console program, press the **CMD** key; when the "COMMAND" prompt appears, type **^E**. This will cause the console screen to display the **exit** command as shown below:

```
COMMAND EXIT
```

Press **RETURN** to exit to the PC's operating system. When you do, you should see the "A:\>" prompt. If you don't see this prompt, just press **RETURN** a few times until you do.

While exiting the console program won't ordinarily affect the computers, it is never a good idea to leave the console program off for very long. Doing so leaves portions of computer activity unrecorded in history, making diagnosis of any problems difficult.

Note that you will also need to exit the console program if you wish to change the console's configuration file—a text file stored on drive A. Editing this file is described in "Console Configuration File" in this section.

Starting the Console

As we mentioned at the beginning of this chapter, the PC must run a program called **console** in order to be able to act as your system's console. Your PC has been set up so that when you turn the power on, it will automatically run the console program.

You can also restart the console software by running the console program yourself. To do this, simply type "console" at the "A:\>" prompt as shown below and press **RETURN**.

```
A:\> console
```

After a few seconds you'll see the console screen partitioned into separate regions for each computer in the system.

Miscellaneous Notes

The console will beep when an error occurs. An error condition may be anything from a modem command that exceeds 80 characters in length to blank spaces in modem passwords. The console will also beep when you reach the bottom of history.

Occasionally, a computer will send a message while you are typing in a command. The computer's message takes precedence over your typing and the console will stop displaying your keystrokes to display the message. However, the console will continue to record what you type. Once it has displayed the message, the console will display everything you typed while the message was being displayed.

×

Final Version

Final Version

Console Configuration File

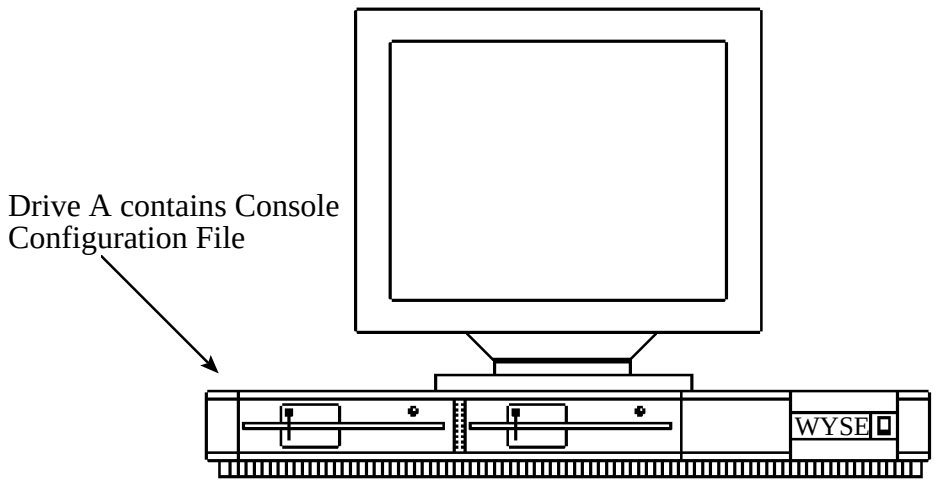
Modifying Console Parameters

As mentioned in “The Console” chapter, your console uses a configuration file to set a number of *parameters*. These parameters include whether disk logging is enabled, information about each of the computers connected to the console, and information about the remote console.

This chapter shows you what these parameters are and how they affect your console’s operation. You’ll also learn how to edit this file in case you ever need to make changes to your console’s operation.

Examining the Configuration File

The configuration file is on your console’s A drive (see below) and is called CONSOLE.CFG. Because it is a text file, you can examine it from the console using the **view** command. (Remember to use **zoom** to display just one region on the screen before using **view**.)



Console

To use **view** to examine the console configuration file, press **CMD** to bring up the “COMMAND” prompt and type “v” followed by “console.cfg” as shown below. When you press **RETURN**, the first line in the console configuration file will be displayed. Continue pressing **RETURN** to scroll through the file.

```
COMMAND v console.cfg
```

By the way, you don’t have to tell **view** to look for this file on drive A because **view** always looks on drive A unless you tell it otherwise.

A sample console configuration file is shown below:

Final Version

```
log b:log 32764
computer                ;"A" computer
name       a
label      WBAS
irq        3
hostess    2c0
speed      9600
;
computer                ;"B" computer
name       b
label      WBAS
irq        3
hostess    2c8
speed      9600
;
computer                ;Archive computer "C"
name       c
label      ARCHIVE
irq        3
hostess    2d0
speed      9600
;
modem                   ;Remote console
password   turtle
timeout    6:00
irq        3
hostess    2d8
speed      1200
```

As you can see, this file is a list of keywords, most of which are followed by parameters. These keywords are described in detail at the end of this chapter.

Two keywords that do not have parameters are “modem” and “computer.” These keywords identify the start of the modem (remote console) section and the computer sections, respectively. Each computer (including the Archive computer, if your system has one) and the modem must have its own section in the console configuration file.

By the way, as is the case with your system’s site files, you can include comments and blank lines in the console configuration file by preceding each comment with a semicolon. The console program will ignore anything following a semicolon.

Editing the Console Configuration File

The only modification you will probably ever need to make to this file is to change the modem password occasionally. So, as an example of how to edit the configuration file, we’ll show you how to do this. In the process, you’ll learn how to use the MS-DOS line editor, **edlin**, to make a few simple changes to this file. For a complete explanation of how to use **edlin**, refer to your MS-DOS manual.

Final Version

To use **edlin**, you must first exit from the console and return to your PC's operating system (MS-DOS). Note that while the console program is not running, no messages from the computers will be displayed on the screen or recorded in history.

To exit from the console, press **CMD** to bring up the "COMMAND" prompt and then press **^E**. This will cause the console to display "EXIT" as shown below:

```
COMMAND EXIT
```

Now press **RETURN** to exit, and you should see the MS-DOS "A:\>" prompt as shown below. If you don't see this prompt, pressing **RETURN** again should display it.

```
A:\>
```

When you see the "A:\>" prompt, use **edlin** to open the configuration file for editing. To do this, type "edlin console.cfg" as shown below and press **RETURN**.

```
A:\> edlin console.cfg
End of input file
*
```

When **edlin** opens CONSOLE.CFG, it will place you at the end of the file (indicated by the "End of input file" message). The first thing you should do is list the file by typing "1p" (the number "1" and the letter "p") and press **RETURN**. This will move you to the first line and list the first page (23 lines) of the file as shown below:

```
A:\> 1p
     1:log b:log 32764
     2:computer              ;"A" computer
     3:name      a
     4:label     WBAS
     5:irq       3
     6:hostess   2c0
     7:speed     9600
     8:;
     9:computer              ;"B" computer
     10:name     b
     11:label    WBAS
     12:irq      3
     13:hostess  2c8
     14:speed    9600
     15:;
     16:computer             ;Archive system "C"
     17:name     c
     18:label    ARCHIVE
     19:irq      3
     20:hostess  2d0
     21:speed    9600
```


Final Version

```
        22:;
        23:modem                ;Remote Console
*
```

After listing one page, **edlin** displays an asterisk prompt on the left indicating that it is waiting for you to enter a command. In this case, the configuration file is longer than 23 lines and cannot fit in one screen. To list the next page, type “p” at the asterisk prompt and press **RETURN**.

```
*p
        24:password    turtle
        25:timeout     6:00
        26:irq         3
        27:hostess     2d8
        28:speed       1200
*
```

As you can see, line 24 contains the modem password. To edit this line we must first move to it. To do this, type “24” (the line you want to move to) and press **RETURN**. **Edlin** will display the contents of this line and, on the next line, display “24:*” to indicate that you can begin typing the new contents of this line.

```
        24:password    turtle
        24:*
```

Now type in the new line as shown below. When you are done, press **RETURN**.

```
        24:password      oreo
*
```

After you press **RETURN**, there should be an asterisk on the left side of the screen as shown above. Once again, **edlin** is waiting for you to enter a command. You are done editing the file, so type “e” for “exit” and press **RETURN** to end the editing session. This saves your changes to the file and returns you to the “A:\>” prompt.

```
* e
A:\>
```

Now start the console program again by typing “console” and pressing **RETURN**.

```
A:\> console
```

Console reads the configuration file as it starts up. If it encounters a keyword that it does not recognize, it will display an error message at the lower left corner and exit back to the “A:\>” prompt. If this happens, you will need to edit the file and fix the error before you will be able to get **console** running.

Final Version

The only other modification you may want to make to the configuration file is the timeout value. The amount of time you enter (in the minutes:seconds format) next to the keyword “timeout” will be the amount of inactivity allowed during a modem connection before the modem will automatically be logged out. (For a full description of “timeout,” see the console configuration keyword definitions below.) If you want to disable the timeout feature, enter a time of “0:00” in the configuration file. You may want to experiment with a few timeout values to find out what is the best value for your system. Once you’ve done that, you can probably leave that value unchanged.

Console Configuration Keywords

As you’ve already seen, each line in the console configuration file (except for comments, which begin with a semicolon) must begin with a valid keyword. Most of these keywords must be followed by a few spaces, or a **TAB** space, and a parameter. Below, we describe each keyword and the kind of parameter that should follow it.

Computer

This keyword indicates the beginning of a computer section and must appear at the top of each computer section in the configuration file.

Hostess <port address>

This keyword is used to indicate which port address (in hex) the console software should use to communicate with a particular device (i.e., a computer or the modem). This information, which is dictated by the hardware, was placed in the configuration file when your system was installed and should not be changed. Each device must have a port address defined in its section.

Irq <interrupt request number>

To get the console’s attention, each device (i.e., the computers and the modem) connected to the console must have its own “interrupt request” with which it can signal the console. The **irq** keyword tells the console which interrupt request to expect from each device.

Each device connected to the console must have this information in its section in the console file. Interrupt request numbers, which are dictated by hardware, were placed in the configuration file when your system was installed and should not be changed.

Label <region name>

This keyword allows you to define a label that the console will use to identify each computer’s region of the console screen. The examples in this manual and the News Operations Manual use the label “WBAS” for the News computers and the label “ARCHIVE” for the Archive computer.

Final Version

The label can be up to fifteen characters long and can contain any alphanumeric character. Since the modem doesn't have a region on the console screen, it is not given a label.

Log <drive:filename> [<computers>] [<max size>]

The **log** keyword allows you to enable disk logging. To do this, follow "log" with the filename you want the console to give the history files. Keep in mind that whatever name you choose, the extensions will always be "A1," A2," "B1," etc. Also, you should precede the filename with the name of the drive where you want the files to be written (usually the B drive).

If you don't follow the filename with a list of the computers for which you want history to be recorded, the console will record history for all computers. However, if you only want history to be recorded for certain computers, follow the file name with the names of those computers.

If you want to specify the size of the disk log files, list the maximum file size (in bytes) after the name of the computers for which you want to record history. If no size is listed, the maximum size of the disk log files will be 16,384 bytes. (Remember that 1K of memory equals 1,024 bytes.)

For example, if you want history for the "A" and "B" computers, but not the "C" computer, to be recorded on the B drive in files with the name "history," you would enable disk logging this way: "log b:history ab." Because no maximum size was entered, the history files will be no more than 16,384 bytes.

How to Determine What Maximum Log Size to Assign

Following is a suggestion on how to determine the maximum log size to be entered with the log keyword.

First, use the DOS **checkdisk** command to determine how much space is available on the disk to be used for disk logging. Then, divide that number by the number of computers whose history you want to log. Next, divide that number by four (the number of history files per computer). The max size should be no greater than the final number you get.

For example, say you determine you have 256K available on the disk. If you want to log the history of three News computers and an archive (for a total of four computers), you would divide 256K by four, giving you 64K. Dividing 64K by four would give you 16K. Therefore, you would set the maximum logging size to 16K or less. Because max size is specified in bytes, a max size of 16K would be entered as "16,384." This happens to be the default value, so if that is the max size you want to use, you can either enter "16,384" with the keyword or not enter any number.

Modem

This keyword indicates the start of a modem section. The section defining the modem must begin with a line containing the word "modem."

Final Version

Name <computer name>
You must use this keyword in each computer's section of the console configuration file to name that computer. Each computer must use "A," "B," "C," or "D" as its name.

Since only one modem can be connected to the console, there is no need to give it a name. Consequently, this keyword is not used in the modem section.

Password <password>
This keyword identifies the modem password that must be entered when someone wants to log in at a remote console. The password can consist of up to eight alphanumeric characters. Unless you set a password, anyone will be able to use a modem to connect to the console—and to your system. This keyword must appear only in the modem section.

Portaddress <port>
This keyword is used to select the DOS address (in hex) the console should use to communicate with a particular device (i.e., a computer or the modem). This information, which is dictated by the hardware, was placed in the configuration file when your system was installed and should not be changed. Each device must have a port address defined in its section. Portaddress is used instead of "hostess" when your system has five computers or four computers and a modem. In that case, the fifth computer (or the modem) must use com1 as its port. The port parameter should always be defined as "3f8."

Speed <baud rate>
This keyword is used to set the baud rate for communication between the console and the computers and modem. You must specify a speed for each computer and the modem. The console's ports communicate at 8 data bits, no parity, one stop bit and one start bit, and Xon/Xoff handshaking.

Timeout <minutes:seconds>
The **timeout** keyword allows you to set a timeout value for any modem connection. The system will automatically log out a modem connection if there is no activity for the specified amount of time. A timeout value of "6:00," for example, would automatically log out a modem connection if there was no activity for six consecutive minutes. This helps when someone finishes working on the console (via a modem) and forgets to log out of the system.

If the timeout value is set to "0:00," the system will not log out the modem connection, regardless of the length of inactivity. If timeout is not included in the configuration file, the system will *default* to a timeout value of "0:00," which disables the feature. The maximum value is 546 minutes and 7 seconds. This keyword should only be used in the modem section.

×

Final Version

Final Version

Console Superuser

Access to Restricted Console Commands

Some console commands are more powerful than others and can be dangerous if used by people who lack the proper training. To allow less-trained staff to perform simple system maintenance while preventing them from using potentially dangerous console commands, the more powerful commands are restricted to *console superusers*.

To become a console superuser, you must know the console superuser password. System Managers, such as yourself, will know this password. You should take great pains to keep it secret from non-System Managers.

Console Superuser vs. Terminal Superuser

There is another kind of superuser called a *terminal superuser*. However, you shouldn't confuse the two. Console superuser status grants you access to certain powerful console commands, while terminal superuser status gives you access to any queue in the database and allows you to create and remove users. (Details on terminal superuser status are available in "User Traits" in Section 4 of the News Operations Manual.)

There are only a few important console commands (such as **utraits** and **dbtraits**) that require that you be a console superuser. In this chapter, we'll show you how to become a console superuser so that you can use these commands and then how to exit from console superuser when you're done. At the end of the chapter, we'll also show you how to change the console superuser password.

Becoming a Console Superuser

To become a console superuser, type "su" (for "superuser") and press **RETURN**. The system will respond by asking you for the console superuser password. Type in the password and press **RETURN** again. To help keep the password secret, the console will not display what you type, so type carefully.

```
WBAS_A: su
password:            (enter password)
SU: so /dev/console
WBAS_A#
```

Once you've entered the password, the console will show that you have become a console superuser by changing the ":" at the end of the *console prompt* to a "#." For instance, the prompt changed to "WBAS_A#" in the example above. If you mistype the password or enter an incorrect password, the console will display a "sorry" message and allow you to try again.

Final Version

Exiting Console Superuser

Once you've become a console superuser, be careful not to leave the console unattended without exiting from console superuser mode. Otherwise, someone else may come along and take advantage of your console superuser status to execute commands they should not use.

To avoid this problem, we recommend that you only become a console superuser when you need to execute a command that requires it. Then, when you're done with that command, exit from console superuser by typing **^D** and pressing **RETURN** as shown below. When you do this, the console prompt will return to normal, indicating that you are no longer a console superuser.

```
WBAS_A# ^D
WBAS_A:
```

The Superuser Password

Since nothing stays secret forever, you should use the **passwd** command to change your system's console superuser password occasionally. As you might expect, you must be a console superuser to use **passwd**, so begin by becoming a console superuser. Then type in "passwd root" and press **RETURN**. By the way, *root* is the UNIX name for superuser and tells the system that you want to change the console superuser password.

```
WBAS_A: su
password:                   (enter password)
SU: so /dev/console
WBAS_A# passwd root
changing password for root
New password:               (enter new password, at least 6 characters)
Verify:                     (enter new password a second time)
WBAS_A# ^D
WBAS_A:
```

The system will tell you that it is "changing password for root" and ask for the new password. To keep your password secret, the console won't display what you type, so type the new password carefully. When you press **RETURN**, the system will ask you to verify (retype) the password so it can make sure you have entered the password as you meant to enter it.

If what you type the second time matches what you originally typed, the system will display the console superuser prompt. If, however, it doesn't match, the system will assume that you made an error and display a "Mismatch - password unchanged" message. If you see this message, you will have to start again.

When you are done, remember to press **^D** to exit from console superuser. Next time you become a console superuser, you will need to use the new password.

Final Version

Choosing a Password

You can choose almost any combination of characters for the new password as long as it is at least six characters long and contains at least one non-numeric character. Generally, you should choose a word that is fairly easy to remember and not longer than ten characters.

Also, the system differentiates between upper and lower case. If you choose a password in upper case, you must enter it in upper case to become a console superuser. Since all the commands you enter at the console must be in lower case, it will be easiest if you choose a password that is all lower case too.

Be sure to keep a record of changes to the console password. Knowing the password is imperative not only for **dbtraits** and other commands, but also for B.A.S. technicians who may require superuser access for problem diagnosis and reconfiguration. Note that if you forget your password, the system may have to be rebooted in a special mode by B.A.S. technicians (on VAX systems) or have its software reloaded (on Wyse Xenix systems).

×

Final Version

Final Version

Section 3 Reference

Final Version

Final Version

Appendix A

Reference Charts

ASCII Control Characters

Decimal Value	How Typed	Code Name	Action
0	CTRL @	<NUL>	
1	CTRL A	<SOH>	
2	CTRL B	<STX>	
3	CTRL C	<ETX>	
4	CTRL D	<EOT>	
5	CTRL E	<ENQ>	
6	CTRL F	<ACK>	
7	CTRL G	<BEL>	Beep
8	CTRL H	<BS>	Same as BACKSPACE key
9	CTRL I	<HT>	Same as TAB key
10	CTRL J	<LF>	Same as LINE FEED key
11	CTRL K	<VT>	
12	CTRL L	<FF>	Top of page on printer
13	CTRL M	<CR>	Same as RETURN or ENTER key
14	CTRL N	<SO>	
15	CTRL O	<SI>	
16	CTRL P	<DLE>	
17	CTRL Q	<DC1>	Undo effect of CTRL S
18	CTRL R	<DC2>	
19	CTRL S	<DC3>	Make computer stop scrolling
20	CTRL T	<DC4>	
21	CTRL U	<NAK>	
22	CTRL V	<SYN>	
23	CTRL W	<ETB>	
24	CTRL X	<CAN>	
25	CTRL Y	<EM>	
26	CTRL Z	<SUB>	
27	CTRL [	<ESC>	Same as ESC key
28	CTRL \	<FS>	
29	CTRL]	<GS>	Call up CONNECT prompt
30	CTRL ^	<RS>	
31	CTRL _	<US>	
32	Space Bar	<SP>	
59	;	<SC>	Semicolon
60	<	<<	Two << needed to generate one <
127	DEL key	<DEL>	

Final Version

Any ASCII decimal value may be substituted for an ASCII code name when placed between ‘<>’s.

Refer to the chapter on “Customizing the Keyboard.”

International Character Descriptions, Sequences, and Values

Note that the “_” character replaces the compose key, <255>, as the first character in each character sequence. Table appears in IBM value order.

Character	Description	Character Sequence	MCS Value	IBM Value
¶	paragraph sign	_p!	182	021
§	section sign	_so	167	022
Ç	C cedilla	_C,	199	128
ü	u umlaut	_u”	252	129
é	e acute	_e’	233	130
â	a circumflex	_a^	226	131
ä	a umlaut	a”	228	132
à	a grave	_a`	224	133
å	a ring	_a*	229	134
ç	c cedilla	_c,	231	135
ê	e circumflex	_e^	234	136
ë	e umlaut	_e”	235	137
è	e grave	_e`	232	138
ï	i umlaut	_i”	207	139
î	i circumflex	_i^	206	140
ì	i grave	_i`	236	141
Ä	A umlaut	_A”	196	142
Å	A ring	_A*	197	143
É	E acute	_E’	201	144
æ	ae ligature	_ae	230	145
Æ	AE ligature	_AE	198	146
ô	o circumflex	_o^	244	147
ö	o umlaut	_o”	246	148
ò	o grave	_o`	242	149
û	u circumflex	_u^	251	150
ù	u grave	_u`	249	151
ÿ	y umlaut	_y”	253	152
Ö	O umlaut	_O”	214	153

Final Version

Ü	U umlaut	_U”	220	154
¢	cent sign	_c/	162	155
£	pound sign	_l=	163	156
¥	yen sign	_y=	165	157
á	a acute	_a’	225	160
í	i acute	_i’	237	161
ó	o acute	_o’	243	162
ú	u acute	_u’	250	163
ñ	n tilde	_n~	241	164
Ñ	N tilde	_N~	209	165
ª	feminine ordinal	_a_	170	166
º	masculine ordinal	_o_	186	167
¿	inverted ?	_??	191	168
½	half fraction	_12	189	171
¼	quarter fraction	_14	188	172
¡	inverted !	_!!	161	173
«	open brackets *	_<<<<	171	174
»	closed brackets	_>>	187	175
ß	sharp s	_ss	223	225
µ	micro sign	_u/	181	230
ø	o slash	_o/	248	237
±	plus or minus sign	_+-	177	241
_	degree sign	_o<sp>	176	248
²	superscript 2	_2^	178	253

* To generate a “<” character, we use “<<” in the keyboard dictionary because a single “<” serves as an introduction to a translation sequence. To generate a “<<” character, therefore, requires the sequence “<<<<.”

Refer to chapters on “Character Mapping” and “Customizing the Keyboard.”

Final Version

International Character Descriptions, Sequences, and Values

Wherein MCS Characters are mapped to approximate IBM Characters

Character MCS/IBM	Description	Character sequence	MCS Value	Substitute IBM Value
Â/â	A circumflex	_A^	194	131
À/à	A grave	_A`	192	133
Ê/ê	E circumflex	_E^	202	136
Ë/ë	E umlaut	_E”	203	137
È/è	E grave	_E`	200	138
Ï/ï	I umlaut	_I”	207	139
Î/î	I circumflex	_I^	206	140
Ì/ì	I grave	_I`	204	141
Ô/ô	O circumflex	_O^	212	147
Ò/ò	O grave	_O`	210	149
Û/û	U circumflex	_U^	219	150
Ù/ù	U grave	_U`	217	151
Ÿ/ÿ	Y umlaut	_Y”	221	152
Á/á	A acute	_A’	193	160
I/í	I acute	_I’	205	161
Ó/ó	O acute	_O’	211	162
Ú/ú	U acute	_U’	218	163
Ø/ø	O slash	_O/	216	237

Refer to chapters on “Character Mapping” and “Customizing the Keyboard” for more information.

Final Version

Correspondence Between Key Labels and Hardware Scan Codes on IBM™-Style Keyboards

Alphanumeric Key Area

Key Label	Hardware Scan Code	Key Label	Hardware Scan Code
A	30	3 or #	04
B	48	4 or $	05
C	46	5 or %	06
D	32	6 or ^	07
E	18	7 or &	08
F	33	8 or *	09
G	34	9 or (	10
H	35	` or ~	41
I	23	- or _	12
J	36	= or +	13
K	37	[or {	26
L	38	] or }	27
M	50	\ or |	43
N	49	; or :	39
O	24	' or "	40
P	25	, or <	51
Q	16	. or >	52
R	19	/ or ?	53
S	31	Shift (left)	42
T	20	Ctrl (left)	29
U	22	Alt (left)	56
V	47	Shift (right)	54
W	17	Ctrl (right)	103
X	45	Alt (right)	102
Y	21	Caps Lock	58
Z	44	Backspace	14
0 or)	11	Tab	15
1 or !	02	Spacebar	57
2 or @	03	Enter	28

Final Version

Numeric/Cursor Keypad

Key Label	Hardware Scan Code
Num Lock	69
/	100
*	55
-	74
+	78
Enter	101
1 or End	79
2 or ↓	80
3 or Pg Dn	81
4 or ←	75
5	76
6 or →	77
7 or Home	71
8 or ↑	72
9 or Pg Up	73
0 or Ins	82
. or Del	83

Function Keys

Key Label	Hardware Scan Code
Esc	1
F1	59
F2	60
F3	61
F4	62
F5	63
F6	64
F7	65
F8	66
F9	67
F10	68
F11	87
F12	88
Print Screen	84
Scroll Lock	70
Pause	89

Final Version

Dedicated Cursor Area and Related Keys

Key Label	Hardware Scan Code
Insert	90
Home	91
Page Up	92
Delete	93
End	94
Page Down	95
Up arrow	96
Left arrow	97
Down arrow	98
Right arrow	99

Refer to chapter on “Customizing the Keyboard” for more information.

Monochrome Attribute Values

	Attribute	Attribute Value
Text:	No Display	0
	Normal	7
	Reverse	112
	Underlined	1
Blinking:	Non-blinking	0
	Blinking	128
Intensify:	Normal	0
	Intensify	8

Refer to chapter on “Video Attributes.”

Final Version

Color Attribute Values

Foreground Color Attributes

Attribute	Attribute Value
Black	0
Blue	1
Green	2
Cyan	3
Red	4
Magenta	5
Brown	6
White	7
Gray	8
Bright Blue	9
Bright Green	10
Bright Cyan	11
Bright Red	12
Bright Magenta	13
Yellow	14
Bright White	15

Background Color Attribute

Attribute	Attribute Value
Black	0
Blue	16
Green	32
Cyan	48
Red	64
Magenta	80
Brown	96
Light Gray	112

Blinking Attribute

Attribute	Attribute Value
Non-blinking	0
Blinking	128

Refer to chapter on “Video Attributes.”

×

Final Version

Appendix B

Computer Keyboards

This section is provided to make it easier for you to customize the keyboard if you are working with one of the following:

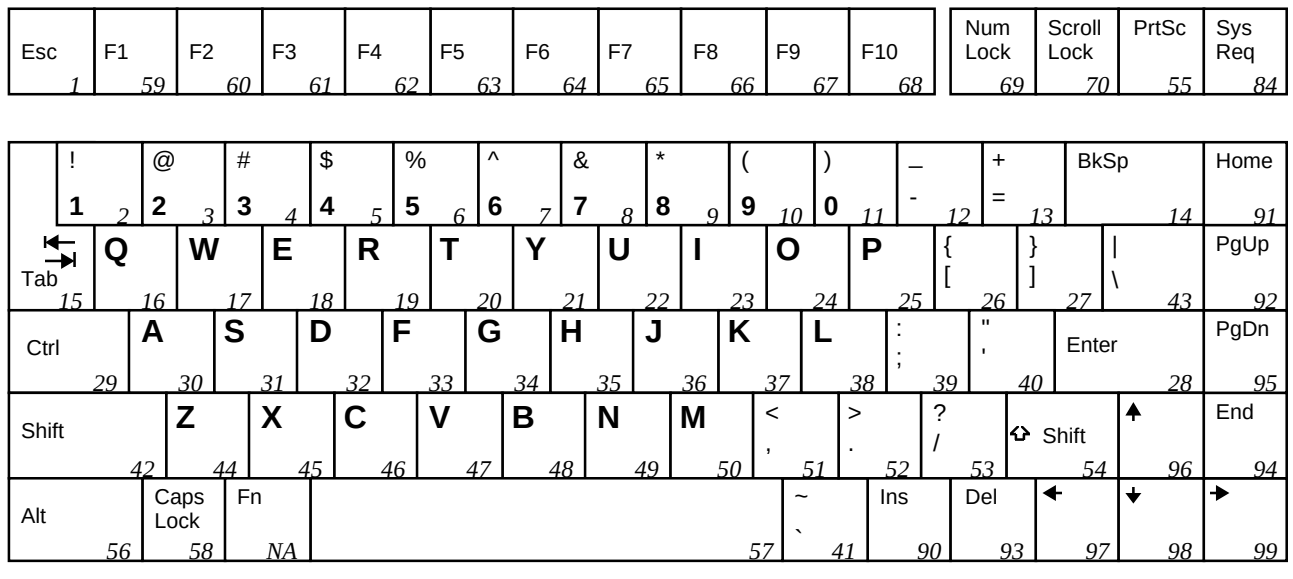
- Toshiba 1200/1000 Laptop Keyboard
- Zenith SUPERSPORT Laptop Keyboard
- IBM Standard Keyboard (XT style)
- IBM Extended Keyboard (AT style)

These are the styles of keyboard most likely to be used with the console program, and they are the keyboards for which we have provided *alternate keyboard dictionaries* for your use. (Refer to the chapter called “Customizing the Keyboard” for more information on using these alternate dictionaries.)

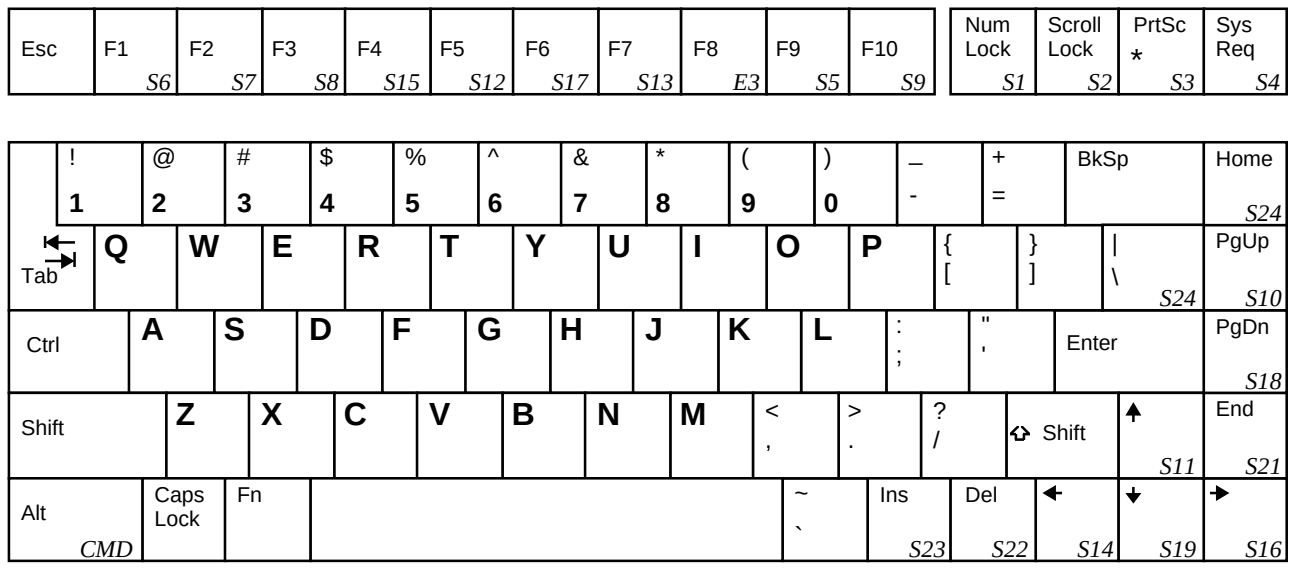
The drawings that appear on the following pages are reproductions of each of these keyboards. Each type of keyboard has at least two drawings. The first drawing, in addition to key labels, has the scan codes printed on each key (they appear in italicized type in the lower right of each key). In the second drawing, we provide in italicized type the function key number that will be assigned to each key when the alternate keyboard dictionary for that computer has been installed. The Toshiba and Zenith computers have two additional drawings, which provide the scan codes and function key numbers, respectively, that result from the use of the **Fn** key on these computers.

Final Version

Toshiba 1200/1000 Keyboard Drawings



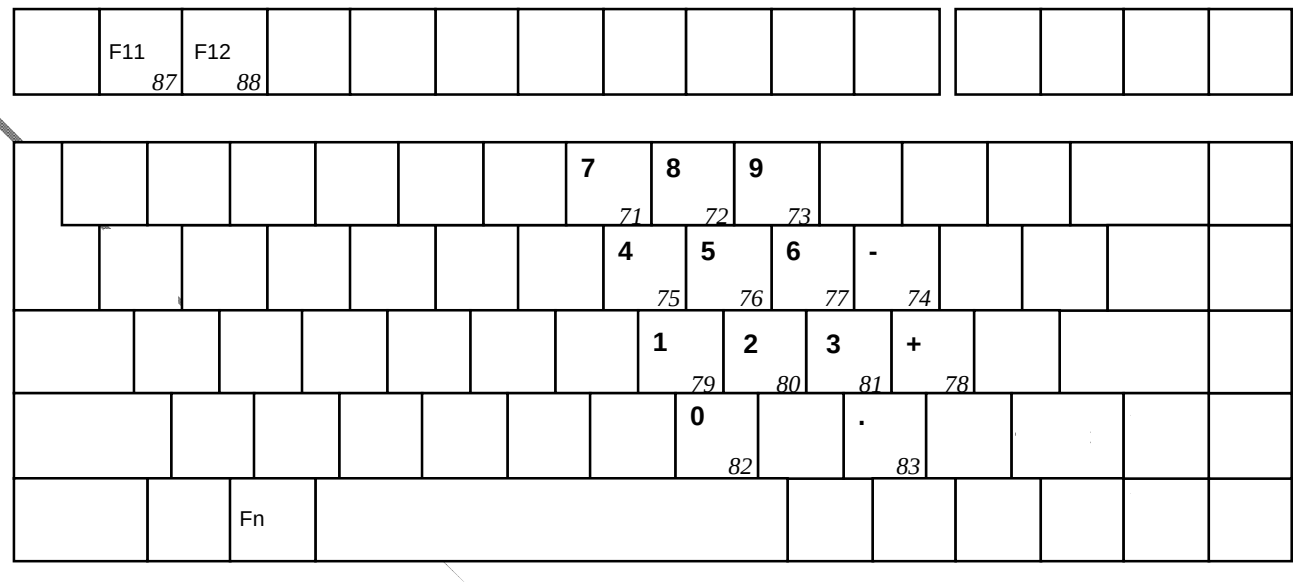
Toshiba 1200/1000 Keyboard Labels and Hardware Scan Codes



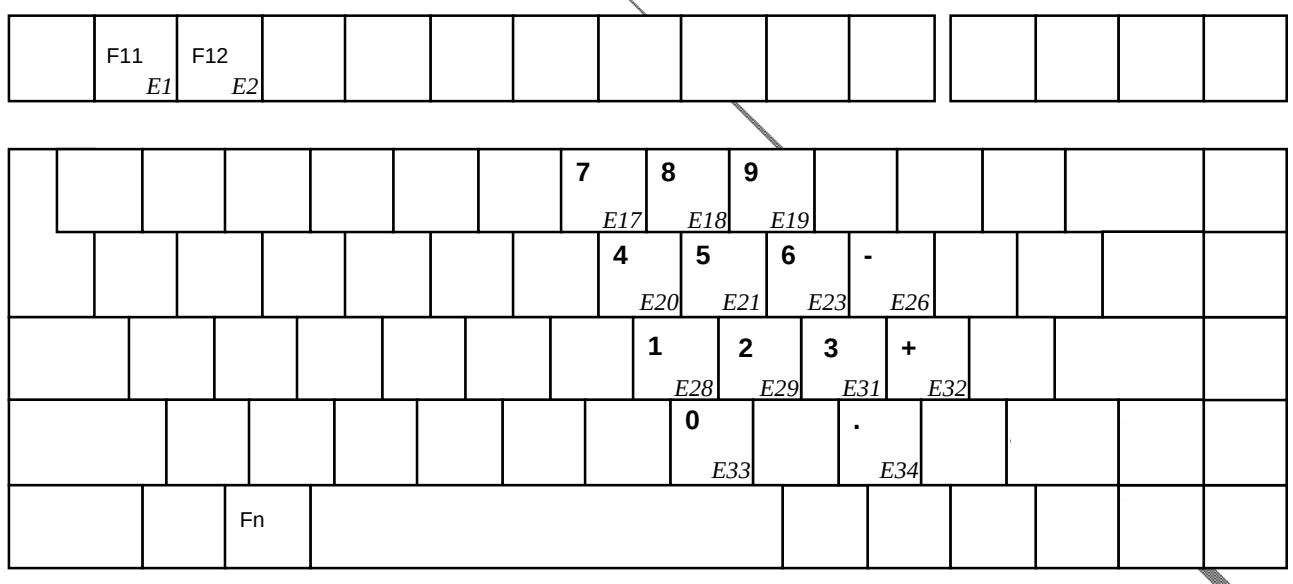
Toshiba 1200/1000 Keyboard Labels and Function Key Numbers

Final Version

Toshiba 1200/1000 Keyboard Drawings



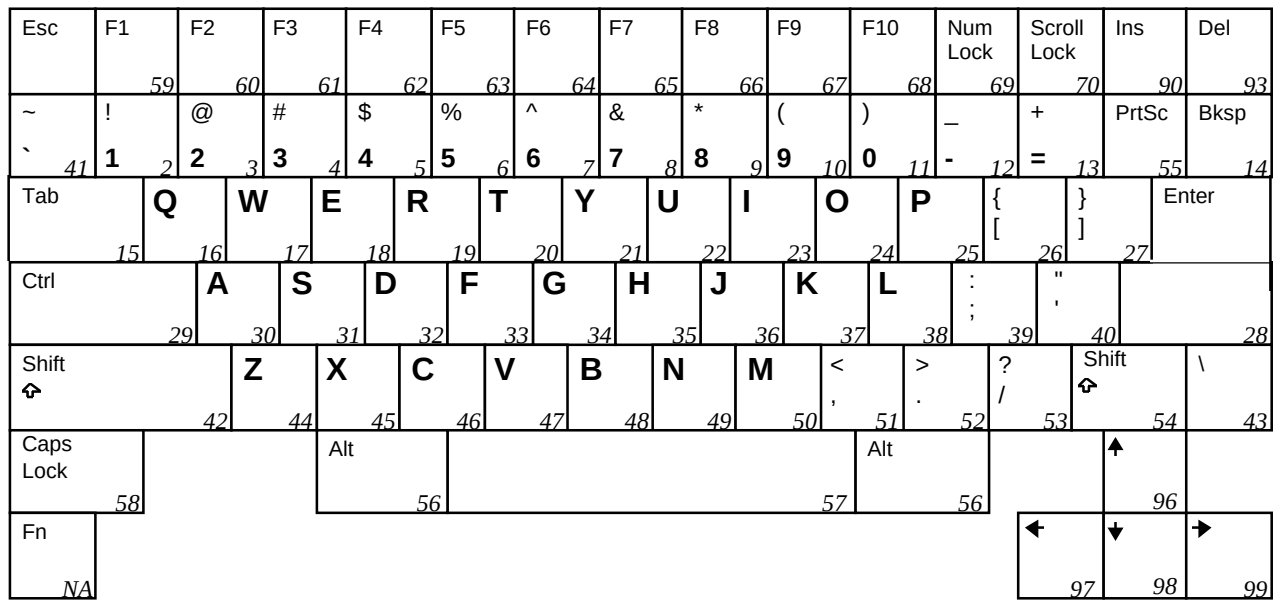
Toshiba 1200/1000 Fn Key Labels and Hardware Scan Codes



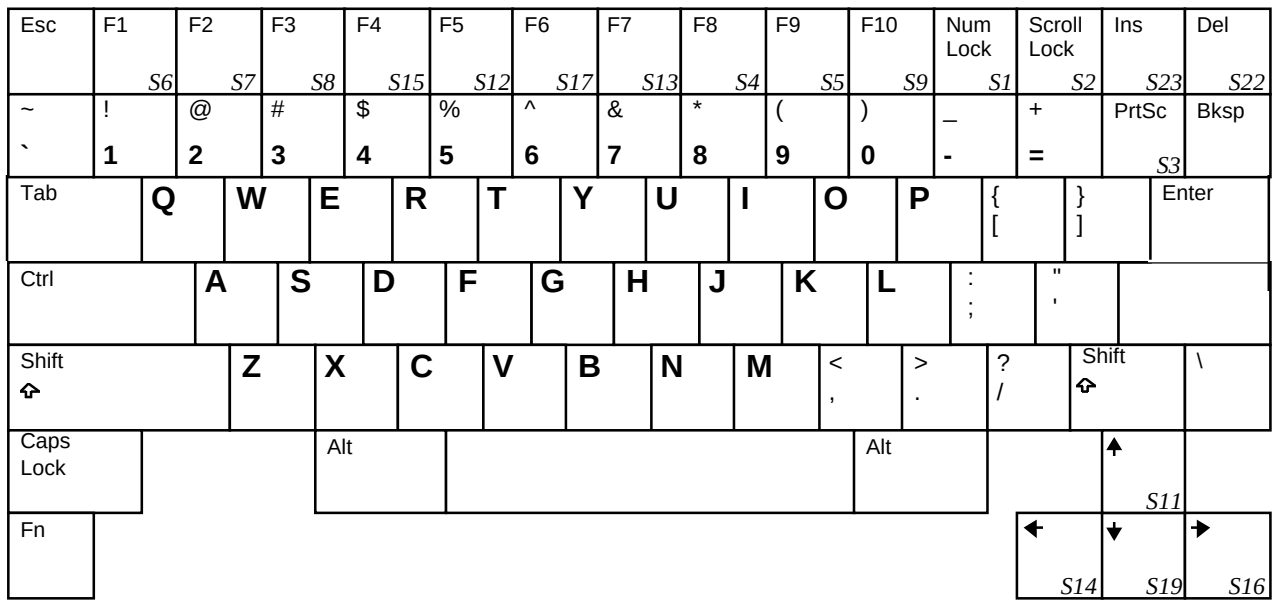
Toshiba 1200/1000 Fn Key Labels and Function Key Numbers

Final Version

Zenith SUPERSPORT Keyboard Drawings



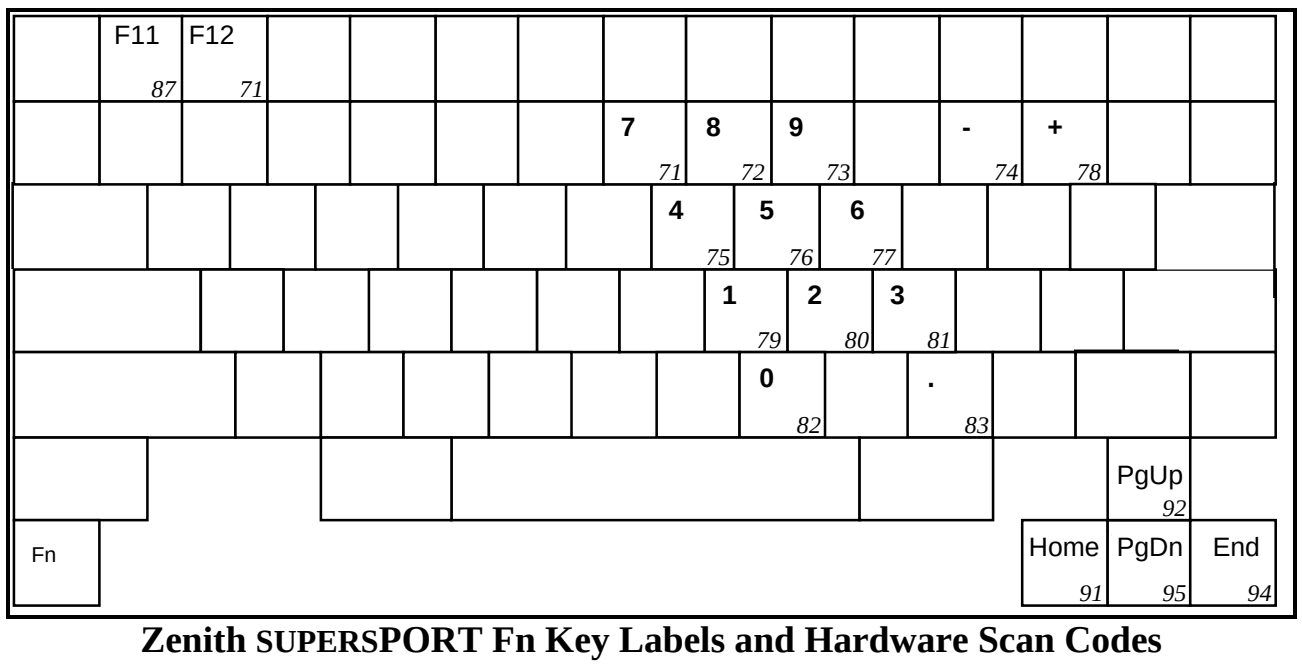
Zenith SUPERSPORT Keyboard Labels and Hardware Scan Codes


Esc F1 S6 F2 S7 F3 S8 F15 S15 F5 S12 F6 S17 F7 S13 F8 S4 F9 S5 F10 S9 Num Lock S1 Scroll Lock S2 Ins S23 Del S22
~ ` 1 ! 2 @ 3 # 4 $ 5 % 6 ^ 7 & 8 * 9 (10) 0 11 _ - 12 + = PrtSc S3 Bksp
Tab Q W E R T Y U I O P { [}] Enter
Ctrl A S D F G H J K L : ; " '
Shift Z X C V B N M < , > . ? / Shift \
Caps Lock Alt Alt S11
Fn S14 S19 S16

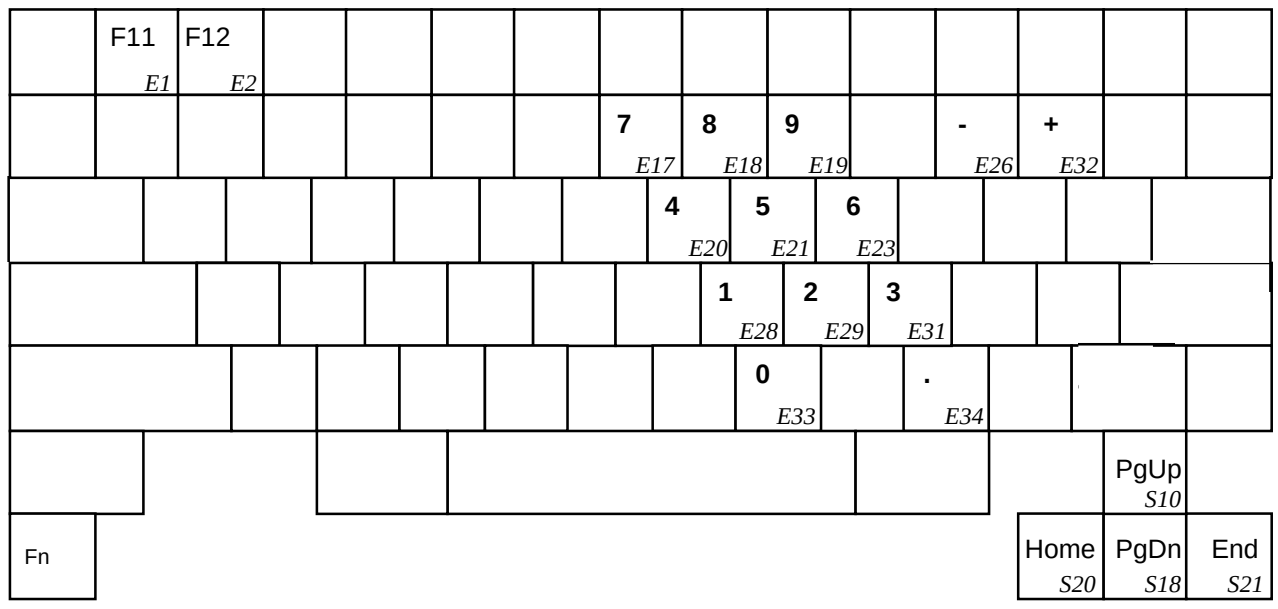
Zenith SUPERSPORT Keyboard Labels and Function Key Numbers

Final Version

Zenith SUPERSPORT Keyboard Drawings

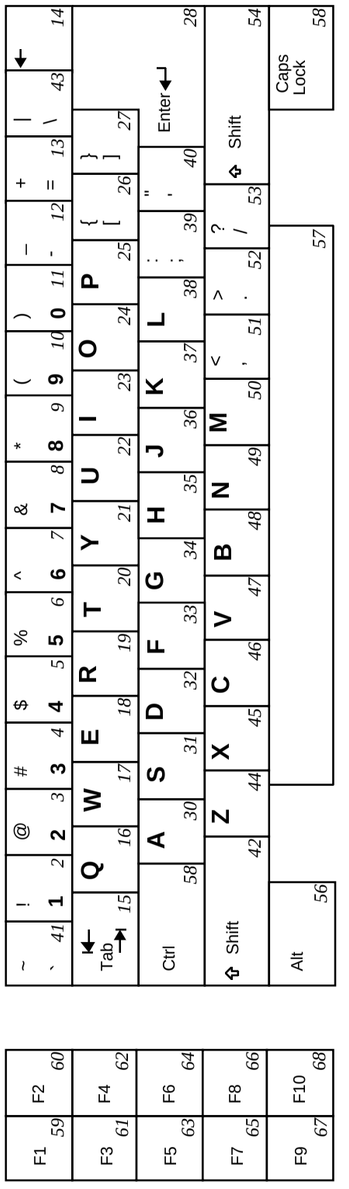


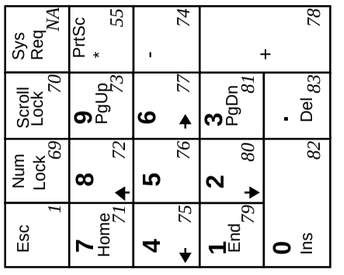
Zenith SUPERSPORT Fn Key Labels and Hardware Scan Codes



Zenith SUPERSPORT Fn Key Labels and Function Key Numbers

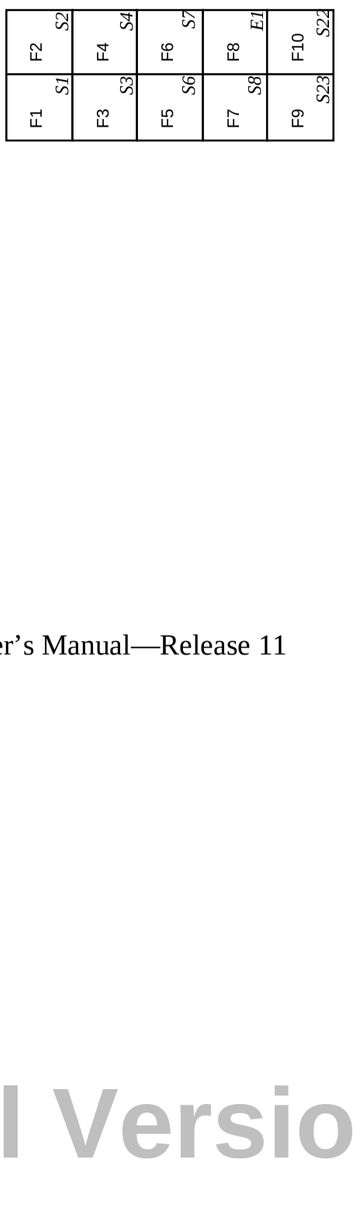
Final Version





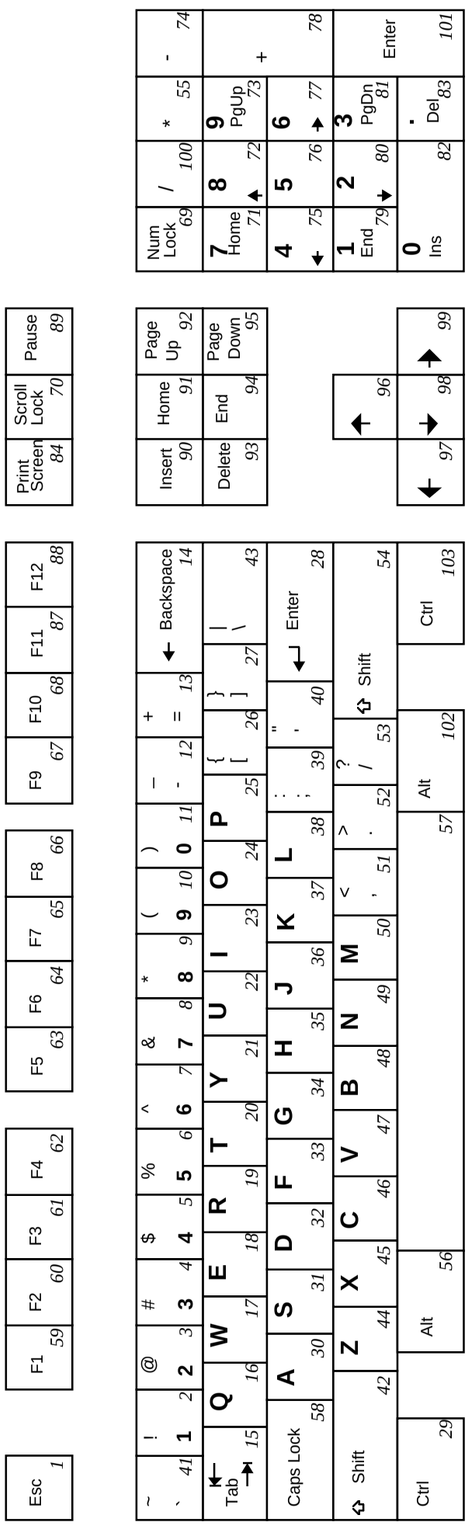
IBM Standard Keyboard Labels and Scan Codes

Final Version



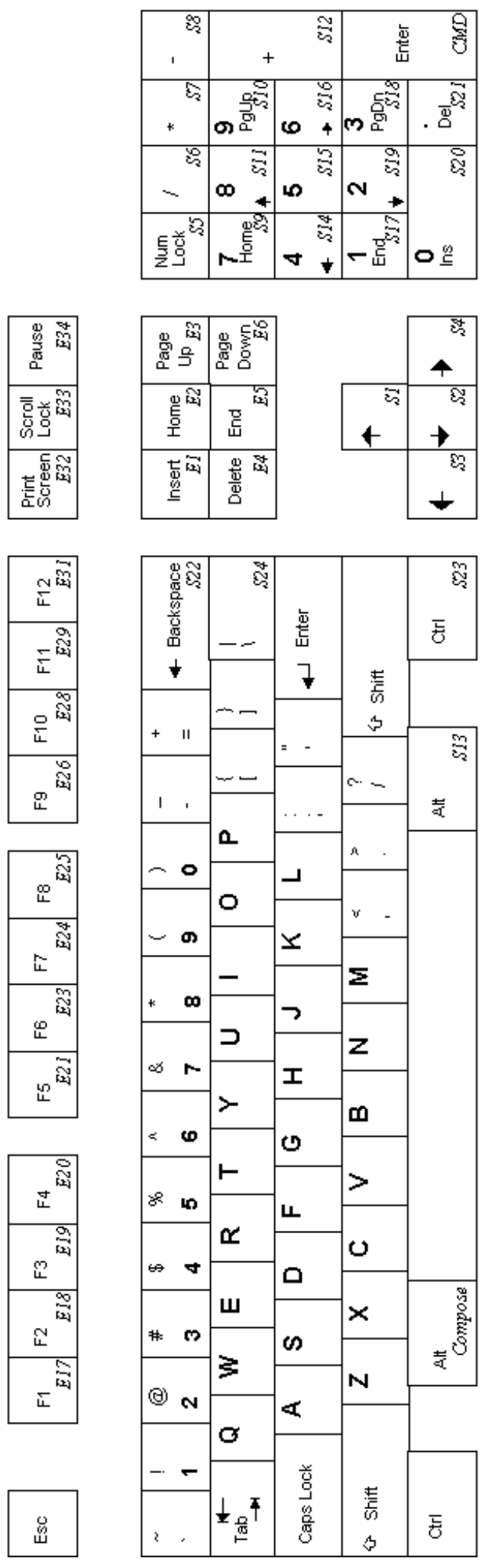
IBM Standard Keyboard Labels and Function Key Numbers

Final Version



IBM Extended Keyboard Labels and Scan Codes

Final Version



IBM Extended Keyboard Labels and Function Key Numbers

Final Version

Final Version

Final Version

Glossary

Alternate Keyboard Dictionaries

A set of *keyboard dictionaries* provided with the console software that are set up to address specific keyboard styles. These include Toshiba 1000/1200, Zenith SUPERSPORT, IMB XT and IBM AT style keyboards. The console software comes with the Toshiba keyboard installed. If you are using one of the other keyboard types, you can install the alternate keyboard dictionary provided for it, and customization of that keyboard will then be greatly simplified. Be sure to check the chapter on "Customizing the Keyboard" and Appendix B in addition to "Setting Up The Console Program" for more information.

ASCII Codes

The standard way to represent characters as numbers, used by most computers and printers. ASCII stands for **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. Along with printable characters, the ASCII codes include control characters to indicate carriage return, backspace, etc. A partial ASCII table can be found in Section 3.

Compose Characters

Characters that do not normally appear on a keyboard (variously known as compose, extended, international, or multinational characters), but which can be assigned to a key or displayed on the screen. See the chapter called "Character Mapping" for more on compose characters. Also see Appendix A.

Compose Key

When the user needs to generate a *compose character* on the screen, they can do so by pressing the compose key, followed by the *compose sequence* that corresponds to the character they desire. The compose key can be defined in the *keyboard dictionary* as "<255>." Check Appendix A for the characters and character sequences that are available.

Compose Sequences

Key combinations that generate characters that do not normally appear on a keyboard (variously known as compose, extended, international or multinational characters), but which can be assigned to a key or displayed on the screen. See the chapter called "Character Mapping" for more on compose sequences. Also see Appendix A.

Configuration File

The configuration file is a text file containing information about the devices connected to the system. This file includes information about the type of device, which port it uses, and how to communicate with the device. The DOS filename for this file is CONSOLE.CFG.

Final Version

Console

An IBM PC-compatible microcomputer that is connected to your system's computers so that you can monitor and maintain your system. The console software can record system messages displayed on the screen and commands entered at the console. It also provides a password-protected modem connection that allows diagnostic and maintenance work to be done from a remote location.

CONSOLE.CFG

This DOS file is your system's configuration file. It includes information about the devices connected to your system. Also see "Configuration File."

Console Command

Any command that you enter at the console. In this manual, all console commands are shown in bold letters (e.g., **dbtraits**).

Console Control Commands

Commands that control the console (as opposed to console commands, which control your system). For example, the "computer" console control command allows you to select a computer, and the "zoom" console control command zooms a computer's window.

Console History

See "History."

CONSOLE.KEY

The DOS file that contains definitions to function keys. The definitions assigned to your console's function keys will remain, even if you turn the console's power off.

Console Prompt

The prompt that appears in each computer's region on the console screen. Each prompt is made up of the system ID and the name of the computer it represents. For example, the console prompt for the "A" computer in a system with the system ID "WBAS" would be "WBAS_A."

Console Superuser

A password-controlled, privileged level of access that allows you to use certain console commands.

DEC VT200 Multinational Character Set

The set of characters defined by DEC that are available on a VT200 terminal for code values between 128 and 255.

Default

A standard feature or value that is automatically assigned by the console program. All defaults can be changed by the System Manager.

Final Version

Dictionary

A DOS file containing information used by the console program. Examples of the types of information contained in the console dictionaries include messages and command abbreviations. You can customize the console software by editing dictionaries and running *makemux*.

Disk Logging

Your console stores the most recent portion of history in memory. Older portions, however, are written out to disk to free up space in memory. The process or writing portions of history out to disk is called *disk logging*.

DOS

Stands for “Disk Operating System.” The operating environment which provides commands for the user to copy, organize, and maintain the console files.

DOS Directory

An area set aside on your hard disk or floppy diskette to contain a certain category of information. For example, the MUX11 directory in DOS contains all the programs and files relating to the console. Also see *sub-directory*.

Edlin

A DOS program used to edit DOS files.

Extended Characters

See “Compose Characters.”

Extension

The “last name” of a DOS file—the three letters following the period in a DOS filename. For example, in the filename “MALTA.TXT,” the extension is “TXT.”

Hardware Scan Code

The number by which a PC identifies each key on its keyboard. Each key generates a unique scan code, even if the same label appears on more than one key. Hardware scan codes are assigned *VT200 character sequences* in the *keyboard dictionary*.

History

A record of system activity maintained by the console. This record contains everything that has been displayed or entered on the console in the recent past. Older portions of the history may be stored in files on the console’s B drive.

IBM Extended Character Set

The set of characters available on a PC running the console program, sometimes abbreviated as IBM. It serves as an adjunct to the standard ASCII character set and includes such characters as £, ¢ and ¥. Each character in the IBM Extended Character Set corresponds to a unique character value as well as a character

Final Version

sequence. IBM values are not the same as *DEC Multinational Character Set* values. MCS values are translated into IBM values, and vice versa, in the videomap dictionary. Also see *compose sequences*.

International Characters

See “Compose Characters.”

Keyboard Dictionary

The DOS file in which *VT200 Character Sequences* are assigned to *Hardware Scan Codes*. The console software comes with several keyboard dictionaries to accommodate various popular computer models.

Literal

Any character the system is to treat literally as text. In general, it is any character *not* meant as a command or some kind of modifier.

Makemux

The program that installs the contents of the console dictionaries into the console program. Whenever a change is made to a dictionary, **makemux** must be run.

Parameter A value that modifies the action of a command. For example, the “t” in a **list t** command is a parameter that instructs the command to list only the active terminals.

Parameters are also used in site files. For example, the values following “terminal” in a terminal configuration line are parameters that define the configuration line for a particular terminal.

Remote Console

When you use a modem to connect to your system’s console from a remote location, you are said to be operating a remote console.

Root

The UNIX name for “superuser.” Normally, you’ll only run into this term when you change the console superuser password.

Select a Computer

Choosing the computer or computers to which you want the console to send the commands you type. For example, if you select the “A” computer, the console will send what you type to that computer.

Shift State

The manner in which a key is pressed—either alone or in combination with one of the following keys: **CAPS LOCK**, **SHIFT**, **CTRL** or **ALT**.

Final Version

Sub-directory

A *DOS directory* that exists “within” another directory.

System Manager

A member of the newsroom staff who has been trained in the routing operation and maintenance of your newsroom system. Console users should be System Managers.

Terminal Superuser

The highest level of security assigned to a user. Terminal superuser status allows a user to access any portion of the database regardless of its restriction. Terminal superuser status should not be confused with *console superuser* status, which applies only to the console.

Translation

The second column of most *dictionaries* contains the description or translation for each standard name. If you want to customize the console program, the translations can be changed.

User

A person working on the B.A.S. News System.

Video Attributes

The special display characteristics that can be applied to text appearing on the PC screen. In the case of a monochrome monitor, these attributes include underlining, boldface, blinking, reverse video, normal text, or certain combinations of these. In the case of a color monitor, these attributes include the color of the characters and the background upon which they appear, as well as whether they blink or not. Console, however, ordinarily only uses normal text. Refer to the chapter called “Video Attributes.”

VT200 Character Sequence

The code used to correspond to each function key on a VT200-style terminal keyboard. Used by the console in the *keyboard dictionary*, where VT200 character sequences are assigned to *hardware scan codes*.

×

Final Version

Final Version

Index

A

ALT key 1- 15
Alternate keyboard dictionaries 3-9, 19
Answer 1- 22
ASCII
  character set 1- 31
  code 1- 17, 3- 19
  control characters 3- 1
Assigning ASCII codes 1- 17
Attribute values 1- 29

B

Background color 1-29
Backup copies 1-3

C

CAPS LOCK key 1-15
Carriage return 1-21
Character mapping 1-31
Character sequences 1-22
  VT200 1-15
CMD key
  console 2-3
  remote console 2-17
Color attribute values 1-28, 3-8
Color monitor 1-6, 28
COLOR.VID dictionary 1-7, 25, 28
“COMMAND” prompt, console 2-3
“COMMAND” prompt, remote console 2-17
Comments, console configuration file 2-22
Compose
  characters 1-21, 31, 3-19
  key 1-22, 3-19
  sequences 1-18, 21, 32, 3-19
Computers
  selecting 2-3
  selecting all 2-4
  selecting using zoom 2-4
  when to select all 2-5
  when to select any 2-5
  when to select master 2-5
CONFIG.SYS file 1-2
Configuration file 2-9, 19, 21, 3-19
  editing 2-22
  viewing 2-21
Console 1-1, 2-1, 3-20
  beeping 2- 19
  changing log disks 2- 12
  CMD key 2-3
  “COMMAND” prompt 2-3
  configuration file 2-21
  configuration file keywords 2-25
  console commands 2-12, 3- 20
  console control commands 2-3, 3- 20
  define function keys 2-14
  disabling disk logging 2-11
  disk logging 2-5
  editing console configuration file 2-22
  enabling disk logging 2-26
  exit to DOS 2-23
  exiting 2-18
  function keys 2-2
  history 2-5
  history commands 2-6
  “HISTORY” prompt 2-8
  keyboard 2-3
  “LOG” message 2-6
  “Log Err” message 2- 11
  logclose command 2-12
  logging out remote console 2-17
  logopen command 2-12
  “MDM” display 2-16
  modem 2-15
  modem command 2-16
  modem connection 1-1, 2-15
  modem open command 2-16
  modem port speed 2-15
  pausing screen display 2-13
  “PRINTER” display 2-7
  printing history 2-7
  reading history in disk files 2-9

Final Version

remote console 1-1
restoring a frozen console 2-18
screen 2-1
setting remote console password 2-27
start up 2-19
starting the program 2-24
temporarily disabling disk logging 2-12
using the view command 2-21
view command 2-10
viewing console configuration file 2-21
viewing recent history 2-10
viewing the console configuration file 2-21
“XOFF” display 2-18
Console command 2-12, 3-20
Console control commands 2-2, 3-20
Console function keys
changing a definition 2-14
including CMD 2-14
including RETURN 2-14
listing a definition 2-15
Console history
bottom 2-6
listing to screen 2-8
pausing output 2-13
top 2-6
viewing recent history 2-6
writing recent history to disk 2-11
Console prompt x, 2-29, 3-20
Console superuser 2-29, 3-20
becoming 2-29
changing the password 2-30
choosing a password 2-31
exiting 2-30
password 2-29
superuser console prompt 2-29
vs. terminal superuser 2-29
CONSOLE.CFG 3- 20
CONSOLE.KEY 3- 20
CTRL key viii, 1-15
Customizing
dictionaries 1-11
keyboard 1-15

D

DEC
character values 1-31
compose sequence 1-21
multinational character set 1-21
VT200 multinational character set 1-31, 3-20
Default 1-26, 2-27, 3-20
Devatt 1-22
Diagnostic modem 1-1
Dictionaries 1-11, 3-21
characters 1-11
installing changes 1-12
keyboard 1-11
opening for editing 1-11
“Table space exceeded” message 1-14
videomap 1-11
Disk logging 2-5, 3-21
determining log size 2-26
disabling 2-11
enabling 2-26
temporarily disabling 2-12
Displayable characters 1-17
DOS v, 3-21
DOS commands
cd 1- 4
checkdisk 2- 26
copy 1- 4
dir 1-6
type 1-15, 25, 31
xcopy 1-3
DOS dictionary 3-21
DOS directory 1-3

E

Editing a dictionary 1-11
Edlin 1-2, 12, 29, 3-21
Enhanced 1-22
Error messages
“LOG ERR” 2-11
“Table space exceeded” 1-14
Escape command 1-34
Exiting
console 2-19
console superuser 2-30
Extension 2-9, 3-21

Final Version

F

Fn key 3-9
Foreground color 1-29
Function key number 3-10
Function keys
  changing a definition 2-14
  console 2-2, 13, 14
  listing a definition 2-15
  remote console 2-17
  selecting computers 2-2

H

Hardware scan codes 1-15, 3-5, 21
  chart 1-19
History 2-5, 3-21
History, console
  older history 2-9
  recent history 2-5
History commands
  bottom 2-6
  down 2-7
  list 2-7
  list, aborting 2-7
  pausing output 2-13
  print 2-8
  top 2-6
  up 2-7
History commands 2- 6
“HISTORY” prompt 2-8
Horizontal tab 1-34

I

IBM
  character values 1-31
  extended character set 1-31, 3-21
  extended keyboard (AT style) 1-8
  extended keyboard drawings 3-16
  standard keyboard (XT style) 1-8
  standard keyboard drawings 3-14
  to MCS character mapping 1-31
Installation 1-3
  upgrade 1-4
Installing changes 1-12
International characters 1-31
International character sequences 3-2
International character values 3-2

K

Key labels 3-5
  chart 1-19
Keyboard
  configuring 1-8
  customizing 1-15
  dictionary 1-8, 11, 15, 3-22
  dictionary descriptions 1-21
Keyboard keys
  CTRL viii

L

Line feed 1-21
Literal 3- 22
“LOG” message 2-6
“LOG ERR” message 2-11
Log size, determining 2-26
Logclose command 2-11
Logging out, remote console 2-17
Logopen command 2-12

M

Makemux 1-6, 7, 3-22
  options 1-12
MCS to IBM character mapping 1-33
Messages
  LOG ERR 2-11
  Table space exceeded 1-14
Modem
  connecting to the console 2-15
  console 1-1, 2-15
  diagnostic 1-1
Modem command 2-16
Modem open command 2-16
Modifier key 1-15
MONO.VID dictionary 1-7, 25
Monochrome attribute values 3-7
Monochrome monitor 1-26
Multinational character set 1-32

Final Version

N

Non-displayable characters 1-34
Non-printable ASCII codes 1-21

P

Parameter 2-21, 3-22
“PRINTER” display 2-7

R

Reference charts 3-1
Release 11 changes vi
Remote console 1-1, 2-15, 3-22
    CMD key 2-17
    COMMAND prompt 2-17
    commands 2-16
    display 2-16
    function keys 2-17
    logging out 2-17
    setting the password 2-27
Requirements for set up 1-2
Root 2-30, 3-22
Running the program 1-9

S

Scan codes 1-15, 3-9
    translation 1-17
Screen, console 2-1
Selecting a computer
    choosing the right computer(s) 2-5
    computer command 2-3
    function keys 2-2
    zoom command 2-4
Selecting a computer 2-3, 3-22
Selecting all computers 2-4
Selecting computers 2-5
Semicolon 1-21
Set up requirements 1-2
SHIFT key 1-15
Shift state 1-16, 3-22
Start up, console 2-19
Starting the program 1-9, 2-24
Sub-directory 1-6, 3-22
System Manager v, 1-1, 3-23

T

“Table space exceeded” error message 1-14
Terminal superuser 2-29, 3-23
    vs. console superuser 2-29
Timeout keyword 2-25, 27
Toshiba 1-2
    keyboard drawings 3-10
    keyboard 1-8
Translation 1-11, 3-23
    MCS and IBM characters 1-31
Type command 1-15, 25, 31

U

Upgrade
    onto dual floppy 1-5
    onto hard drive 1-4
    onto single floppy 1-5
User 1-1, 3-23

V

Video attributes 1-25, 3-8, 23
    combining 1-26
Video display, configuring 1-7
Videomap dictionary 1-11, 31
VIDEOMAP.DCT
    dictionary 1-7
    file 1-31
View command, console 2-10
VT100 terminal screen 1-26
VT200
    character sequences 1-15, 3-23
    terminal screen 1-26

X

“XOFF” display 2-18

Z

Zenith supersPORT™ 1-8
    keyboard drawings 3-12
Zoom command 2-4

×

Final Version